VISIT…

Second Edition

HIGH DYNAMIC RANGE IMAGING

Acquisition, Display and Image-Based Lighting

Erik Reinhard • Greg Ward • Sumanta Pattanaik
Paul Debevec • Wolfgang Heidrich • Karol Myszkowski

MORGAN KAUFMANN

HIGH DYNAMIC RANGE IMAGING

This page intentionally left blank

HIGH DYNAMIC RANGE IMAGING

ACQUISITION, DISPLAY, AND IMAGE-BASED LIGHTING

Second Edition

ERIK REINHARD
SUMANTA PATTANAIK
WOLFGANG HEIDRICH
GREG WARD
PAUL DEBEVEC
KAROL MYSZKOWSKI

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO
Morgan Kaufmann Publisher is an imprint of Elsevier

Morgan Kaufmann Publishers is an imprint of Elsevier.
30 Corporate Drive, Suite 400, Burlington, MA 01803, USA

This book is printed on acid-free paper.

Copyright © 2010, Elsevier Inc. All rights reserved.

Greg Ward: photograph
Tania Pouli: design

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher. Details on how to seek permission, further information about the Publisher's permissions policies and our arrangements with organizations such as the Copyright Clearance Center and the Copyright Licensing Agency, can be found at our website: *www.elsevier.com/permissions*.

This book and the individual contributions contained in it are protected under copyright by the Publisher (other than as may be noted herein).

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

Library of Congress Cataloging-in-Publication Data
High dynamic range imaging : acquisition, display, and image-based lighting / Erik Reinhard ... [et al.]. – 2nd ed.
p. cm. – (The Morgan Kaufmann series in computer graphics)
Includes bibliographical references and index.
ISBN 978-0-12-374914-7 (alk. paper)
1. High dynamic range imaging. 2. Photography–Digital techniques. 3. Photography–Printing processes. 4. Image processing. 5. Electroluminescent display systems. 6. Computer graphics. I. Reinhard, Erik, 1968–
TR594.H54 2010
771–dc22
2010005628

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

ISBN: 978-0-12-374914-7

For information on all Morgan Kaufmann publications, visit our Web site at *www.mkp.com* or *www.elsevierdirect.com*

Typeset by: diacriTech, India

Printed in China
10 11 12 13 5 4 3 2 1

Working together to grow libraries in developing countries
www.elsevier.com | www.bookaid.org | www.sabre.org
ELSEVIER BOOK AID International Sabre Foundation

To Tania. You make all the difference.
— E.R.

To my daughters Alina and Tessala, my wife Elizabeth,
my step-daughters Melanie and Sophia, and artists everywhere.
— G.W.

To my family.
— S.P.

To my parents, who showed me the magic of the darkroom.
— P.D.

To my family.
— W.H.

To Beata, Joanna and Adam.
— K.M.

This page intentionally left blank

Contents

PREFACE TO THE SECOND EDITION xv

PREFACE TO THE FIRST EDITION xvii

01 INTRODUCTION 1

02 LIGHT AND COLOR 19

2.1 Radiometry 19
2.2 Photometry 25
2.3 Colorimetry 28
2.4 Color Spaces 34
2.5 White Point and Illuminants 36
2.6 Spectral Sharpening 47
2.7 Color Opponent Spaces 48
2.7.1 CIELUV 56
2.7.2 CIELAB 58
2.7.3 IPT 59
2.8 Color Correction 61
2.9 Color Appearance 61
2.9.1 CIECAM02 66
2.9.2 Refinements 72
2.10 Display Gamma 77
2.11 Brightness Encoding 80
2.12 Standard RGB Color Spaces 82

03 HIGH DYNAMIC RANGE IMAGE ENCODINGS 91

3.1 Low- versus High Dynamic Range Encodings 91
3.2 Applications of HDR Images 93
3.3 HDR Image Formats 95
3.3.1 The HDR Format 96
3.3.2 The TIFF Float and LogLuv Formats 98
3.3.3 The OpenEXR Format 103
3.3.4 Other Encodings 104
3.3.5 "Lossy" HDR Formats 105
3.3.6 Microsoft's HD Photo Format 110
3.4 HDR Encoding Comparison 110
3.5 Conclusions 117

04 HIGH DYNAMIC RANGE VIDEO ENCODINGS 119

4.1 Custom HDR Video Coding 123
4.1.1 Color Space for Lossy HDR Pixel Encoding 123
4.1.2 MPEG-4 Extension to HDR Video 130
4.1.3 An HDR Video Player 133
4.2 Backwards-Compatible HDR Video Compression 136
4.2.1 HDR MPEG-4 Visual 137
4.2.2 Scalable Bit-depth Coding in the H.264/AVC Framework 141

05 HDR IMAGE CAPTURE 145

5.1 Photography and Light Measurement 145
5.1.1 Camera RAW 147
5.2 HDR Image Capture from Multiple Exposures 148
5.3 Film Scanning 151
5.4 Image Registration/Alignment 154
5.5 The Median Threshold Bitmap Alignment Technique 155
5.5.1 Threshold Noise 161
5.5.2 Overall Algorithm 164

5.5.3 Efficiency Considerations 167
5.5.4 Results 168
5.5.5 Adding Rotational Alignment 169
5.6 Other Alignment Methods 170
5.7 Deriving the Camera Response Function 171
5.7.1 Debevec and Malik Technique 171
5.7.2 Mitsunaga and Nayar Technique 174
5.7.3 Choosing Image Samples for Response Recovery 177
5.7.4 Deriving Sample Values from Histograms 181
5.7.5 Caveats and Calibration 182
5.8 Noise Removal 183
5.9 Ghost Removal 185
5.10 Lens Flare Removal 188
5.10.1 The Point Spread Function 190
5.10.2 Estimating the PSF 193
5.10.3 Removing the PSF 196
5.11 HDR Capture Hardware 198
5.11.1 Capture of Multiple Exposures 198
5.11.2 Viper FilmStream 200
5.11.3 Panavision 201
5.11.4 Pixim 201
5.11.5 SpheronVR 202
5.11.6 Weiss AG 202
5.12 Conclusion 203

06 DISPLAY DEVICES AND PRINTING TECHNOLOGIES 205

6.1 Display Technologies 205
6.1.1 Cathode Ray Tubes 206
6.1.2 Plasma Displays 208
6.1.3 Liquid Crystal Displays 209
6.1.4 Reflective Display Technologies 213

6.1.5 Projection Technologies 214
6.1.6 Emerging Display Technologies 215
6.2 Local-Dimming HDR Displays 216
6.2.1 Local-Dimming Television Screens 219
6.2.2 Dual Modulation in Large-Format Projection 220
6.2.3 Image Processing for Dual-Modulation Screens 221
6.3 Printing 224
6.3.1 The Reflection Print 225
6.3.2 Transparent Media 227
6.3.3 Ward's HDR Still Image Viewer 228
6.4 Conclusions 230

07 PERCEPTION-BASED TONE REPRODUCTION 233

7.1 Tone-Mapping Problem 233
7.2 Human Visual Adaptation 237
7.2.1 The Pupil 239
7.2.2 The Rod and Cone Systems 240
7.2.3 Photopigment Depletion and Regeneration 243
7.2.4 Photoreceptor Mechanisms 243
7.3 Visual Adaptation Models for HDR Tone Mapping 252
7.3.1 Photoreceptor Adaptation Model for Tone Mapping 252
7.3.2 The TVI Model for Tone Mapping 255
7.4 Background Intensity in Complex Images 256
7.4.1 Image Average as I_b 256
7.4.2 Local Average as I_b 257
7.4.3 Multiscale Adaptation 260
7.5 Dynamics of Visual Adaptation 262
7.6 Design Considerations 264
7.6.1 Calibration 265
7.6.2 Color Images 268
7.6.3 Forward and Backward Models 270
7.7 Summary 276

08 TONE-REPRODUCTION OPERATORS 277

8.1 Sigmoidal Tone-Reproduction Operators 277
8.1.1 Photoreceptor Model 277
8.1.2 Photographic Tone Reproduction 285
8.1.3 Local Sigmoidal Compression 291
8.1.4 Adaptive Gain Control 294
8.2 Image Appearance Models 297
8.2.1 Multiscale Observer Model 297
8.2.2 iCAM 306
8.2.3 iCAM06 312
8.2.4 Color Appearance under Extended Luminance Levels 317
8.3 Other HVS-Based Models 323
8.3.1 Brightness-Preserving Operator 324
8.3.2 Retinex 329
8.3.3 Ashikhmin's Model 331
8.3.4 Lightness Perception 338
8.3.5 Subband Encoding 340
8.3.6 Display-Adaptive Tone Reproduction 344
8.4 Apparent Contrast and Brightness Enhancement 349
8.4.1 Cornsweet Illusion 352
8.4.2 Glare Illusion 367
8.5 Other Tone-Reproduction Operators 376
8.5.1 Histogram Adjustment 377
8.5.2 Bilateral Filtering 380
8.5.3 Gradient Domain Compression 391
8.5.4 Multiscale Optimization Frameworks 396
8.6 Exposure Fusion 400

09 INVERSE TONE REPRODUCTION 405

9.1 Expansion Functions 406
9.1.1 Expanding Highlights 406
9.1.2 Inverse Photographic Operator 408

9.1.3 Gamma Expansion 410
9.1.4 LDR2HDR 412
9.2 Under- and Overexposed Material 415
9.2.1 Hallucination 416
9.2.2 Video Processing 418
9.3 Suppressing Quantization and Encoding Artifacts 422
9.4 Preference Studies 425
9.4.1 Well-Exposed Images 425
9.4.2 Under- and Overexposed Images 428
9.5 Suggested Applications 432
9.6 Summary 434

10 VISIBLE DIFFERENCE PREDICTORS 435
10.1 Subjective versus Objective Quality Metrics 436
10.2 Classification of Objective Quality Metrics 438
10.3 FR Quality Metrics 438
10.4 Pixel-Based Metrics 440
10.4.1 Perceptually Uniform Spaces for Pixel Intensity Encoding 442
10.4.2 JND-Scaled Space 444
10.5 SSIM Index 448
10.6 Perception-Based Fidelity Metrics 451
10.6.1 Multichannel Models 453
10.6.2 Multichannel HDR Image-Quality Metrics 454
10.7 The HDR Visible Differences Predictor 455
10.7.1 Eye Optics Modeling 458
10.7.2 Amplitude Nonlinearity Modeling 460
10.7.3 Contrast Sensitivity Function 464
10.7.4 Cortex Transform 468
10.7.5 Visual Masking and Phase Uncertainty 472
10.7.6 Psychometric Function and Probability Summation 477

10.8 Dynamic Range-Independent Image-Quality Metric 479
10.8.1 Applications 483
10.8.2 Comparison with SSIM and HDR VDP 486
10.9 Suprathreshold HDR Image-Quality Metrics 487
10.9.1 Wilson's Transducer 489
10.9.2 Mantiuk's Transducer 491
10.9.3 A Metric Based on Wilson's Transducer 493
10.10 Accounting for Partial Adaptation 494
10.11 Summary 497

11 IMAGE-BASED LIGHTING 501
11.1 Basic IBL 504
11.1.1 Acquire and Assemble the Light Probe 504
11.1.2 Model the Geometry and Reflectance of the Scene 506
11.1.3 Map the Light Probe to an Emissive Surface Surrounding the Scene 506
11.1.4 Render the Scene as Illuminated by the IBL Environment 508
11.1.5 Postprocess the Renderings 508
11.2 Capturing Light Probe Images 515
11.2.1 Photographing a Mirrored Sphere 516
11.2.2 Tiled Photographs 520
11.2.3 Fish-Eye Lenses 523
11.2.4 Scanning Panoramic Cameras 523
11.2.5 Capturing Environments with Very Bright Sources 525
11.3 Omnidirectional Image Mappings 535
11.3.1 Ideal Mirrored Sphere 535
11.3.2 Angular Map 537
11.3.3 Latitude–Longitude 538
11.3.4 Cube Map 540
11.4 How a Global Illumination Renderer Computes IBL Images 543
11.4.1 Sampling Other Surface Reflectance Types 547

11.5 Sampling Incident Illumination Efficiently 549
11.5.1 Identifying Light Sources 553
11.5.2 Converting a Light Probe into a Constellation of Light Sources 561
11.5.3 Importance Sampling 569
11.6 Simulating Shadows and Scene-Object Interreflection 572
11.6.1 Differential Rendering 576
11.6.2 Rendering into a Nondiffuse Local Scene 577
11.7 Useful IBL Approximations 579
11.7.1 Environment Mapping 579
11.7.2 Ambient Occlusion 583
11.8 Image-Based Lighting for Real Objects and People 586
11.8.1 A Technique for Lighting Real Subjects 587
11.8.2 Relighting from Compressed Image Data Sets 588
11.9 Conclusions 593

APPENDIX A: LIST OF SYMBOLS 595

BIBLIOGRAPHY 599

INDEX 641

Preface to the Second Edition

After the final draft of a manuscript is handed to the publisher, a large amount of work goes on. First, the book is copyedited and then typeset, proofs are created and vetted, and then the printer gets to produce the actual copies. About three quarters of a year passes between the delivery of the manuscript and the first copies appearing in shops and online.

The first edition of this book appeared late 2005. At that time, the first pieces of essential research on topics in high dynamic range imaging had already started to appear; before the book came out, and after we sent the final draft to the editor. This is testament to the fact that high dynamic range imaging is a fast moving field indeed.

Although the first edition has served its purpose well, it soon became clear that an update would eventually become necessary. Not only has new work appeared on well established topics such as tone reproduction but also several new topics have emerged that consequently were not covered at all in the first edition. These include inverse tone reproduction, visible difference predictors, and HDR video encodings. We have added a chapter for each of these to this second edition. We also welcome Wolfgang Heidrich and Karol Myszkowski to the author team. They have done a tremendous job creating new chapters and have also helped update the manuscript throughout.

A thorough restructuring of the tone reproduction chapters has also become necessary to account for new work in this area, to provide an exposition that is less an enumeration of existing operators, and to offer a better exposition of the underlying thoughts and ideas. Updates to all other chapters have made the book up-to-date as of this writing. Most images have been replaced with better or more visually appealing examples.

We expect that between delivery of the final manuscript and you opening this book for the first time, new research will have advanced the field of high dynamic

range imaging once more. In the mean time, we hope that this work may serve as a reference for this exciting discipline called high dynamic range imaging.

ACKNOWLEDGMENTS TO THE SECOND EDITION

In addition to the collection of people who have helped us in some way for the first edition, we would like to thank in no particular order the following colleagues, students, friends, and family for their help and support:

Helge Seetzen, Gerwin Damberg, Akiko Yoshida, Kaleigh Smith, Rafał Mantiuk, Grzegorz Krawczyk, Tunç Ozan Aydın, Scott Daly, Tobias Ritschel, Piotr Didyk, Thorsten Grosch, Elmar Eisemann, Robert Herzog, Hans-Peter Seidel, Mateusz Malinowski, Vlastimil Havran, Daniel Brosch, Kristina Scherbaum, Michael Schöberl, Alexander Oberdörster, Dawid Pająk, Radosław Mantiuk, Anna Tomaszewska, Diego Gutierrez, Douglas Cunningham, Mike Fraser, Sriram Subramanian, Nishan Canagarajah, Ben Long, Dave Coulthurst, Timo Kunkel, Craig Todd, Brent Wilson, Anne Webster, Lorne Whitehead, Holly Rushmeier, John Peterson, Kadi Bouatouch, Xavier Pueyo, Gustavo Patow, and Jorge Lopez.

Ian Ashdown and Joe Geigel have found the time and the energy to review an early draft of this book. This is the second time they have helped us a great deal, and once again they have given us a large collection of comments, which we have taken to heart as well as we could. Thank you for your efforts! Sections of the book have been proofread by Martin Cadik, Tunç Ozan Aydın, Rafał Mantiuk, and Tania Pouli, for which we are extremely grateful. This book was produced under the care of Gregory Chalson and Heather Scherer at Morgan Kaufmann Publishers.

In regard to Chapter 11, we wish to thank Andrew Jones, Chris Tchou, Andrew Gardner, Tim Hawkins, Andreas Wenger, Per Einarsson, Maya Martinez, David Wertheimer, Bill Swartout, and Lora Chen for their help during the preparation of this chapter. The Parthenon model was created with the additional collaboration of Brian Emerson, Marc Brownlow, Philippe Martinez, Jessi Stumpfel, Marcos Fajardo, Therese Lundgren, Nathan Yun, CNR-Pisa, Alias, and Geometry Systems. Portions of the work described in this chapter have been sponsored by a National Science Foundation Graduate Research Fellowship, a MURI Initiative on 3D direct visualization from ONR and BMDO (grant FDN00014-96-1-1200), Interval Research Corporation, the University of Southern California, and the U.S. Army Research, Development, and Engineering Command (RDECOM).

Preface to the First Edition

The thought of writing this book began with the realization that not a single book existed with the title "HDR? Duh!" While we rejected the idea of this title shortly after, both the idea for this book and its title matured, and now you have the final result in your hands.

High dynamic range imaging is an emerging field and for good reasons. You are either already convinced about that, or we hope to convince you with this book. At the same time, research in this area is an amazing amount of fun, and we hope that some of that shines through as well.

Together, the four authors are active in pretty much all areas of high dynamic range imaging, including capture devices, display devices, file formats, dynamic range reduction, and image-based lighting. This book recounts our experience with these topics. It exists in the hope that you find it useful in some sense.

The visual quality of high dynamic range images is vastly higher than conventional low dynamic range images. The difference is as big as the difference between black-and-white and color television. Once the technology matures, high dynamic range imaging becomes the norm rather than the exception. It not only affects people in specialized fields such as film and photography, computer graphics, and lighting design, but it affects everybody who works with images. It would be fair to say that this is effectively everybody.

High dynamic range imaging is already gaining widespread acceptance in the film industry, photography, and computer graphics. Other fields will follow soon. In all likelihood, general acceptance happens as soon as high dynamic range display devices are available for the mass market. The prognosis is that this may be as little as only a few years away.

At the time of writing, there exists no single source of information that can be used both as the basis for a course on high dynamic range imaging and as a work of reference. With a burgeoning market for high dynamic range imaging, we offer

this book as a source of information for all aspects of high dynamic range imaging, including image capture, storage, manipulation, and display.

ACKNOWLEDGMENTS TO THE FIRST EDITION

This book would be unimaginable without the help of a vast number of colleagues, friends, and family. In random order, we gratefully acknowledge their help and support:

Colleagues, friends, and family who have contributed to this book in one form or another: Peter Shirley, Erum Arif Khan, Ahmet Oğuz Akyüz, Grzegorz Krawczyk, Karol Myszkowski, James Ferwerda, Jack Tumblin, Frédo Durand, Prasun Choudhury, Raanan Fattal, Dani Lischinski, Frédéric Drago, Kate Devlin, Michael Ashikhmin, Michael Stark, Mark Fairchild, Garrett Johnson, Karen Louden, Ed Chang, Kristi Potter, Franz and Ineke Reinhard, Bruce and Amy Gooch, Aaron and Karen Lefohn, Nan Schaller, Walt Bankes, Kirt Witte, William B. Thompson, Charles Hughes, Chris Stapleton, Greg Downing, Maryann Simmons, Helge Seetzen, Heinrich Bülthoff, Alan Chalmers, Rod Bogart, Florian Kainz, Drew Hess, Chris Cox, Dan Baum, Martin Newell, Neil McPhail, Richard MacKellar, Mehlika Inanici, Paul Nolan, Brian Wandell, Alex Lindsay, Greg Durrett, Lisa Yimm, Hector Yee, Sam Leffler, and Marc Fontonyot.

Extremely helpful were the comments of the reviewers who ploughed through early drafts of this book: Ian Ashdown, Matt Pharr, Charles Poynton, Brian Smits, Joe Geigel, Josh Anon, Matthew Trentacoste, and the anonymous reviewers.

Several institutes, organizations, and companies have given us their kind support. The Albin Polasek museum (http://www.polasek.org) allowed us to monopolize one of their beautiful galleries to take high dynamic range images. Several of these are shown throughout the book. The Color and Vision Research Laboratories at the Institute of Ophthalmology, UCL, has an extremely helpful publicly available online resource with color related data sets (http://cvrl.ioo.ucl.ac.uk). SMaL cameras have given us a prototype HDR security camera that gave us useful insights (http://www.smalcamera.com). idRuna donated a copy of their Photogenics software — a high dynamic range image editing program (http://www.idruna.com).

Last but not least we thank Tim Cox, Richard Camp, and Brian Barsky at Morgan Kaufmann for the outstanding job they have done producing this book.

Introduction

01

There are many applications that involve digital images. They are created with modern digital cameras and scanners, rendered with advanced computer graphics techniques, or produced with drawing programs. These days, most applications rely on graphical representations of some kind.

During their lifetime, digital images undergo a number of transformations. First they are created using one of the above techniques. Then, they are stored on some digital medium, possibly edited with various image-processing techniques, and ultimately either displayed on a computer monitor or printed on hard copy.

Currently, there is a trend toward producing and using higher-resolution images. For example, at the time of writing digital cameras routinely boast up to 12 megapixel (Mpixel) sensors, with 20 Mpixel sensors available. Digital scanning-backs offer resolutions that are substantially higher. For illustrative purposes, the effect of different image resolutions on the visual quality of an image is shown in Figure 1.1.

Although the trend toward higher-resolution images is apparent, there is a major shift in thinking about digital images that pertains to the range of values that each pixel may represent. Currently, the vast majority of color images are represented with a byte per pixel for each of the red, green, and blue channels. With three bytes per pixel, about 16.7 million different colors can be assigned to each pixel. This is known in many software packages as "millions of colors."

This may seem an impressively large number at first, but it should be noted that there are still only 256 values for each of the red, green, and blue components of each pixel. Having just 256 values per color channel is inadequate to represent

© 2010, Elsevier Inc. All rights reserved.
DOI: 10.1016/B978-0-12-374914-7.00001-4

FIGURE 1.1 *Increasing the number of pixels in an image reduces aliasing artifacts. The image on the left has a resolution of* 163 *by* 246 *pixels, whereas the image on the right has a resolution of* 1632 *by* 2464.

many scenes. Some examples are given in Figure 1.2, where the best exposures of a sequence of autobracketed shots are shown on the left. In these scenes, we have both light and dark areas in the same scene. As a result, the sky shown in the background is overexposed. At the same time, the foreground is too dark.

The same figure shows on the right examples that were created, stored, and prepared for printing with techniques discussed in this book. In other words, these are high dynamic range (HDR) images, which were subsequently prepared for printing.

FIGURE 1.2 *Optimally exposed conventional images (left) versus images created with techniques described in this book (right).*

Here, the exposure of both the indoors as well as the outdoors areas has improved. While these images show more detail in both the dark and light areas, this is despite the fact that this image is shown on paper so that the range of values seen in this image is not higher than in a conventional image. Thus, even in the absence of a display device capable of displaying HDR images, there are advantages to using HDR imaging. The differences between the left and right images in Figure 1.2 would be significantly larger if they were displayed on a display device such as those discussed in Chapter 6.

The term "dynamic range" requires some clarification. It is a dimensionless quantity that can be used to refer to several different physical measures [66]. For images, it is the ratio between the lightest and darkest pixel. Arguably, the minimum and maximum value of an image are by definition outliers, so this measure of dynamic range is not necessarily robust. However, a more reliable measure of dynamic range can be obtained by excluding a percentage of the lightest and darkest pixels.

For a display, the dynamic range is the ratio of the maximum and minimum luminance that it is capable of emitting. The dynamic range of a camera is the ratio of the luminance that just saturates the sensor and the luminance that lifts the camera response to one standard deviation above the noise floor [66].

The range of values afforded by a conventional image is around two orders of magnitude, stored as a byte for each of the red, green, and blue channels per pixel. It is not possible to directly print images with a much higher dynamic range. Thus, to simulate the effect of reducing an HDR image to within a displayable range, we reduce a conventional photograph in dynamic range to well below two orders of magnitude. As an example, Figure 1.3 shows a low dynamic range (LDR) image (8 bits per color channel per pixel), and the same image reduced to only 4 bits per color channel per pixel. Thus, fewer bits means a lower visual quality. Although for some scenes 8 bits per color channel is enough, there are countless situations where 8 bits are not enough.

One of the reasons for this is that the real world produces a much larger range than the two orders of magnitude common in current digital imaging. For instance, the sun at noon may be 100 million times lighter than starlight [91,304]. Typical ambient luminance levels for commonly encountered scenes are given in Table 1.1.[1]

1 Luminance, defined in the following chapter, is a measure of how light a scene appears.

FIGURE 1.3 *The image on the left is represented with a bit depth of 4 bits. The image on the right is represented with 8 bits per color channel.*

The human visual system is able to adapt to lighting conditions that vary by nearly 10 orders of magnitude [91]. Within a scene, the human visual system functions over a range of around five orders of magnitude simultaneously.

This is in stark contrast with typical cathode ray tube (CRT) displays, which are able to reproduce around two orders of magnitude of intensity variation. Their limitation lies in the fact that phosphors cannot be excited beyond a given limit. For this reason, 8-bit D/A converters are traditionally sufficient to generate analog display signals. Higher bit depths are usually not used because the display

Condition	Illumination (cd/m^2)
Starlight	10^{-3}
Moonlight	10^{-1}
Indoor lighting	10^{2}
Sunlight	10^{5}
Maximum intensity of common CRT monitors	10^{2}

TABLE 1.1 *Ambient luminance levels for some common lighting environments (from Wandell's book* Foundations of Vision *[334]).*

would not be able to reproduce such images at levels that are practical for human viewing.[2]

A similar concept holds for typical modern liquid crystal displays (LCD). Their operating range is limited by the strength of the backlight. Although LCDs tend to be somewhat brighter than CRT displays, their brightness is not orders of magnitude larger. A new trend is that energy-saving displays are introduced into the market. Here, an array of individually addressable light-emitting diode (LED) backlights is used to reduce the output of LEDs in screen regions that depict darker material. Commercial systems use this to reduce energy consumption, although the dynamic range is improved as a side effect.

Because current display devices are not able to reproduce a range of luminances anywhere near the capability of the human visual system, images are typically encoded with a byte per color channel per pixel. This encoding normally happens when the image is captured. This is a less than optimal situation because much of the information available in a scene is irretrievably lost at capture time.

A much preferable approach would be to capture the scene with a range of intensities and level of quantization that is representative of the scene, rather than any subsequent display device. Alternatively, images should at least contain a range of

2 It would be possible to reproduce a much larger set of values on CRT displays at levels too low for humans to perceive.

values that is matched to the limits of human vision. All relevant information may then be retained until the moment that the image needs to be displayed on a display device that is not able to reproduce this range of intensities. This includes current CRT, LCD, and plasma devices, as well as all printed media.

Images that store a depiction of the scene in a range of intensities commensurate with the scene are called HDR or alternatively "radiance maps." However, images suitable for display with current display technology are called LDR.

This book is specifically about HDR images. These images are not inherently different from LDR images, but there are many implications regarding the creation, storage, use, and display of such images. There are also many opportunities for creative use of HDR images that would otherwise be beyond our reach.

Just as there are clear advantages of using high-resolution images, there are also major advantages of using HDR data. HDR images and video are matched to the scenes they depict, rather than the display devices on which they are meant to be displayed. As a result, the fidelity of HDR images is much higher than that of conventional images. This benefits most image processing that may be applied during the lifetime of an image.

As an example, correcting the white balance of an LDR image may be difficult because of the presence of overexposed pixels, a problem that exists to a lesser extent with properly captured HDR images. This important issue is discussed in Section 2.8. HDR imaging also allows creative color manipulation and better captures highly saturated colors, as shown in Figure 1.4. It is also less important to light the scene carefully with light coming from behind the photographer, as demonstrated in Figure 1.5. Other image postprocessing tasks that become easier with the use of HDR data include color, contrast, and brightness adjustments. Such tasks may scale pixel values nonlinearly such that parts of the range of values require a higher precision than can be accommodated by traditional 8-bit pixel encodings. An HDR image representation would reduce precision errors to below humanly detectable levels.

In addition, if light in a scene can be accurately represented with an HDR image, such images may be effectively used in rendering applications. In particular, HDR images may be used as complex light sources that light conventionally modeled 3D geometry. The lighting effects thus obtained would be extremely difficult to model in any other way. This application is discussed in detail in Chapter 11.

FIGURE 1.4 *Color manipulation achieved on an HDR capture (left) produced the image on the right. The left HDR image was captured under normal daylight (overcast sky). The right image shows a color transformation achieved with the algorithm detailed in Section 8.1.1.*

Furthermore, there is a trend toward better display devices. The first prototypes of HDR display devices have been around for several years now [288,287]. Examples are shown in Figure 1.6. A more recent Dolby prototype is shown displaying an HDR image in Figure 1.7. For comparison, a conventional display is showing a tone-mapped image of the same scene in this figure. Commercial HDR displays are now available from SIM2 in the form of their Solar47 PRO displays.

In general, the availability of HDR displays will create a much larger market for HDR imaging. Because LDR images will look no better on HDR display devices than they do on conventional display devices, there will be an increasing demand for technology that can capture, store, and manipulate HDR data directly. In the

FIGURE 1.5 *Photographing images with very high contrasts is easier with HDR imaging. The left image shows a conventional photograph, whereas the right image was created using HDR techniques. (Image courtesy of Tania Pouli)*

meantime, there is a push toward salvaging legacy content for display on HDR display devices. Techniques to increase the dynamic range of conventional LDR imagery are discussed in Chapter 9.

Conversely, it is entirely possible to prepare an HDR image for display on an LDR display device, as shown in Figure 1.2. In fact, it is much easier to remove information (i.e., dynamic range) from an image than it is to expand the dynamic range of imagery. This is because of quantization artifacts that may become visible, as well as regions that may be under- or overexposed. It is, therefore, common sense to create and store imagery in an HDR format, even if HDR display devices are ultimately not used to display them. Such considerations should play an important role, for instance, in the design of digital heritage and cultural archival systems.

Another fact that is difficult to convey on printed media is that HDR images and video, when properly displayed on an HDR display device, simply look gorgeous! The difference between HDR display and conventional imaging is easily as big a step

FIGURE 1.6 *Sunnybrook Technologies (left) and BrightSide (right) prototype HDR display devices.*

forward as the transition from black-and-white to color television. For this reason alone, HDR imaging is becoming the norm rather than the exception, and it was certainly one of the reasons to write this book.

The technology required to create, store, manipulate, and display HDR images is currently still emerging. However, there is already a substantial body of research available on HDR imaging, which we collect and catalog in this book. Topics addressed in this book are as follows:

Light and Color. HDR imaging borrows ideas from several fields that study light and color. The following chapter reviews several concepts from radiometry, photometry, and color appearance, and forms the background for the remainder of the book.

HDR Image Encoding. Once HDR data is acquired, it needs to be stored in some fashion. There are currently a few different HDR file formats emerging. The design considerations for HDR file formats include the size of the resulting files, the total range that may be represented (i.e., the ratio between the largest representable number and the smallest), and the smallest step size between successive values. These trade-offs are discussed in Chapter 3, which also introduces standards for HDR image storage.

FIGURE 1.7 *Dolby prototype HDR display, together with a conventional display showing a tone-mapped version of the same image.*

HDR Video Encoding. Recently, encoding standards for high dynamic video have become available. These standards require backward compatibility with existing (MPEG) standards, while still encoding a much higher dynamic range. The issues involved in HDR video encoding are discussed in Chapter 4.

HDR Image Capture. HDR images may be created in two fundamentally different ways. The first method uses rendering algorithms and other computer graphics techniques. The second method uses conventional (LDR) photo cameras to capture HDR data. This may be achieved by photographing a static scene multiple times where for each frame the exposure time is varied. This leads to a sequence of images, which may be combined into a single HDR image. An example is shown in Figure 1.8, and this technique is explained in detail in Chapter 5.

This approach, generally, requires the subject matter to remain still between shots, and the camera should also be placed on a tripod. This limits the range of photographs that may be taken. Fortunately, several techniques exist that align images, remove ghosts, and reduce the effect of lens flare, thus expanding the range of HDR photographs that may be created. These techniques are all discussed in Chapter 5.

In addition, in due time, photo, film, and video cameras that are able to directly capture HDR data will become available. As an example, the FilmStream Viper is a digital camera that directly captures HDR data. Although an impressive system, its main drawback is that it produces raw image data at such a phenomenal rate that hard drive storage tends to fill up rather quickly. This is perhaps less a problem in the studio, where bulky storage facilities may be available, but the use of such a camera on location is restricted by the limited capacity of portable hard drives. It directly highlights the need for efficient file formats to store HDR video. Storage issues are discussed further in Chapter 3.

HDR security cameras are, now, also available. The main argument for using HDR capturing techniques for security applications is that typical locations are entrances to buildings. Conventional video cameras are, typically, not able to capture faithfully the interior of a building at the same time that the exterior is monitored through the window. An HDR camera would be able to simultaneously record indoor as well as outdoor activities.

FIGURE 1.8 *Multiple exposures (shown on the left) may be combined into one HDR image (right).*

Many consumer-level photo cameras are equipped with up to 12-bit A/D converters and make this extra resolution available through proprietary RAW[3] formats (see Chapter 3). However, 10–12 bits of linear data afford about the same

3 Cameras are often allow images to be saved in a proprietary RAW format, which bypasses most of the firmware processing, and therefore comes closest to what the sensor has captured. Each manufacturer uses its own format, which can even vary per camera model.

precision as an 8-bit γ-compressed format, and may therefore still be considered LDR. Some professional cameras are now equipped with 14-bit A/D converters.

HDR Display Devices. Just as display devices have driven the use of 8-bit image processing for the last 20 years, the advent of HDR display devices will impact the general acceptance of HDR imaging technology.

Current proposals for HDR display devices typically use a form of back-projection, either in a system that resembles a slide viewer or in technology that replaces the single backlight of an LCD with a low-resolution but HDR-projective system. The latter technology, thus, provides HDR display capabilities by means of a projector or LED array that lights the LCD from behind with a spatially varying light pattern [288,287]. The display augments this projected image with a high-resolution, but LDR, LCD. These emerging display technologies are presented in Chapter 6.

Dynamic Range Reduction. Although HDR display technology will become generally available in the near future, it will take time before most users have made the transition. At the same time, printed media will never become HDR, as this would entail the invention of light-emitting paper. As a result, there will always be a need to prepare HDR imagery for display on LDR devices.

It is generally recognized that linear scaling followed by quantization to 8 bits per channel per pixel will produce a displayable image that looks nothing like the original scene. It is, therefore, important to preserve somehow key qualities of HDR images when preparing them for display. The process of reducing the range of values in an HDR image such that the result becomes displayable in some meaningful way is called "dynamic range reduction." Specific algorithms that achieve dynamic range reduction are referred to as "tone-mapping operators" or "tone-reproduction operators." The display of a tone-mapped image should perceptually match the depicted scene (Figure 1.9).

As dynamic range reduction requires preservation of certain scene characteristics, it is important to study how humans perceive scenes and images. Many tone-reproduction algorithms rely wholly or in part on some insights of human vision, not in the least because the human visual system solves a similar dynamic range reduction problem in a seemingly effortless manner. We survey current knowledge of the human visual system as it applies to HDR imaging, and in particular to dynamic range reduction, in Chapter 7.

FIGURE 1.9 *The monitor displays a tone-mapped HDR image depicting the background. A good tone-reproduction operator would cause the scene and the displayed image to appear the same to a human observer, in this case giving the impression of being a transparent frame.*

Tone Reproduction. Although there are many algorithms capable of mapping HDR images to an LDR display device, there are only a handful of fundamentally different classes of algorithms. Chapters 7 and 8 present an overview of most currently known algorithms and discuss their advantages and disadvantages. Many sequences of images are provided for each operator that show how parameter settings affect the image appearance.

Inverse Tone Reproduction. With almost all digital media currently available in LDR format, and HDR display devices available on the market, there is a clear need to display

LDR legacy content on HDR display devices. Algorithms to achieve upscaling in a perceptually plausible manner are discussed in Chapter 9.

Visible Difference Predictors. For various purposes, it is important to be able to compare images with one another, and to assess how perceptually different the two images are. Such techniques often rely on implicit assumptions, such as a given 8-bit range of values, and the use of integers. To determine if two images, or regions within images, are perceptually distinguishable therefore becomes a much more difficult problem if these assumptions are removed. In HDR imaging, quantization is effectively removed, and the highest value in each image can be different and bears little relation to the minimum and mean of the image. To overcome these issues, new metrics to determine visible differences between images have become available. These are discussed in Chapter 10.

Image-Based Lighting. In Chapter 11, we explore in detail one particular application of HDR imaging, namely, image-based lighting. Computer graphics is generally concerned with the creation of images by means of simulating how light bounces through a scene [104,177,292]. In many cases, geometric primitives such as points, triangles, polygons, and splines are used to model a scene. These are then annotated with material specifications, which describe how light interacts with these surfaces. In addition, light sources need to be specified to determine how the scene is lit. All this information is then fed to a rendering algorithm, which simulates light and produces an image. Well-known examples are given by films such as the *Shrek* and *Toy Story* series.

A recent development in rendering realistic scenes takes images as primitives. Traditionally, images are used as textures to describe how a surface varies over space. As surface reflectance ranges between reflecting 1% and 99% of all incoming light, the ability of a diffuse surface to reflect light is inherently LDR. It is, therefore, perfectly acceptable to use LDR images to describe things such as woodgrain on tables or the pattern of reflectance of a gravel path. However, surfaces that reflect light specularly may cause highlights that have nearly the same luminance as the light sources that they reflect. In such cases, materials need to be represented with a much higher precision.

In addition, images may be used as complex sources of light within otherwise conventional rendering algorithms [61], as shown in Figure 1.10. Here, we cannot

FIGURE 1.10 *HDR images may be used to light an artificial scene.*

get away with using LDR data because the range of light emitted by various parts of a scene is much larger than the two orders of magnitude available with conventional imaging. If we were to light an artificial scene with a representation of a real scene, we have to resort to capturing this real scene in HDR. This example of HDR image usage is described in detail in Chapter 11.

Although the concept of HDR imaging is straightforward, namely, representing scenes with values commensurate to real-world light levels, the implications to all aspects of imaging are profound. In this book, opportunities and challenges with respect to HDR image acquisition, storage, processing, and display are cataloged in the hope that this contributes to the general acceptance of this exciting technology.

This page intentionally left blank

Light and Color

02

The emerging field of high dynamic range (HDR) imaging is directly linked to diverse existing disciplines such as radiometry, photometry, colorimetry, and color appearance — each dealing with specific aspects of light and its perception by humans [11]. In this chapter, we discuss all aspects of color that are relevant to HDR imaging. This chapter is intended to provide background information that forms the basis of later chapters.

2.1 RADIOMETRY

The term "scene" indicates either an artificial environment or a real environment that may become the topic of an image. Such environments are occupied with objects that each have the capability to reflect light. The ability of materials to reflect light is called "reflectance."

Radiometry is the science concerned with measuring light. This section first briefly summarizes some of the quantities that may be measured and their units. Then, properties of light and how they relate to digital imaging are discussed.

Light is radiant energy and is measured in Joules. Since light propagates through media such as space, air, and water, we are interested in derived quantities that measure how light propagates. These include radiant energy integrated over time, space, or angle. The definitions of these quantities and their units are given in Table 2.1 and should be interpreted as follows.

© 2010, Elsevier Inc. All rights reserved.
DOI: 10.1016/B978-0-12-374914-7.00002-6

Quantity	Unit	Definition
Radiant energy (Q_e)	J (Joule)	
Radiant power (P_e)	$J/s = W$ (Watt)	$P = \frac{dQ}{dt}$
Radiant exitance (M_e)	W/m^2	$M = \frac{dP}{dA}$
Irradiance (E_e)	W/m^2	$E = \frac{dP}{dA}$
Radiant intensity (I_e)	W/sr	$I = \frac{dP}{d\omega}$
Radiance (L_e)	$W/m^2/sr$	$L = \frac{d^2P}{dA \cos\theta \, d\omega}$

TABLE 2.1 *Radiometric quantities. The cosine term in the definition of L_e is the angle between the surface normal and the angle of incidence, as shown in Figure 2.4. Other quantities are shown in Figures 2.1 through 2.3.*

Since light travels through space, the flow of radiant energy may be measured. It is indicated with radiant power or radiant flux and is measured in Joules per second or Watts. It is thus a measure of energy per unit of time.

Radiant flux density is the radiant flux per unit area and is known as "irradiance" if we are interested in flux arriving from all possible directions at a point on a surface (Figure 2.1) and as "radiant exitance" for flux leaving a point on a surface in all possible directions (Figure 2.2). Both irradiance and radiant exitance are measured in Watts per square meter. These are, therefore, measures of energy per unit of time and per unit of area.

If we consider an infinitesimally small point light source, the light emitted into a particular direction is called "radiant intensity" and is measured in Watts per steradian (Figure 2.3). A steradian is a measure of solid angle corresponding to area on the unit sphere. Radiant intensity thus measures energy per unit of time per unit of direction.

FIGURE 2.1 Irradiance: power incident upon unit area $\mathrm{d}A$.

FIGURE 2.2 Radiant exitance: power emitted per unit area.

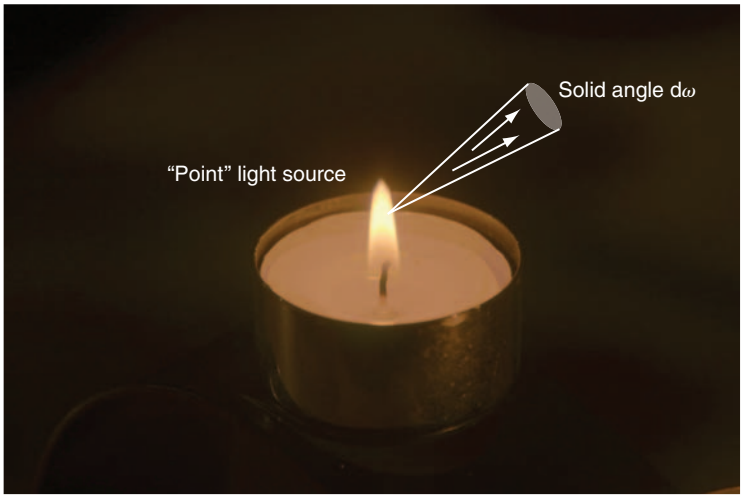


FIGURE 2.3 *Radiant intensity: power per solid angle* $\mathrm{d}\omega$.

Flux passing through, leaving, or arriving at a point in a particular direction is known as "radiance" and is measured in Watts per square meter per steradian (Figure 2.4). It is a measure of energy per unit of time as well as per unit of area and per unit of direction. Light that hits a point on a surface from a particular direction is at the heart of image formation. For instance, the combination of shutter, lens, and sensor in a (digital) camera restricts incoming light in this fashion.

When a picture is taken, the shutter is open for a small amount of time. During that time, light is focused through a lens, thus limiting the number of directions from which light is received. The image sensor is partitioned into small pixels so that each pixel records light over a small area. The light recorded by a pixel may be modeled by the "measurement equation" — see, for example, [159] for details. Since a camera records radiance, it is, therefore, possible to relate the voltages extracted

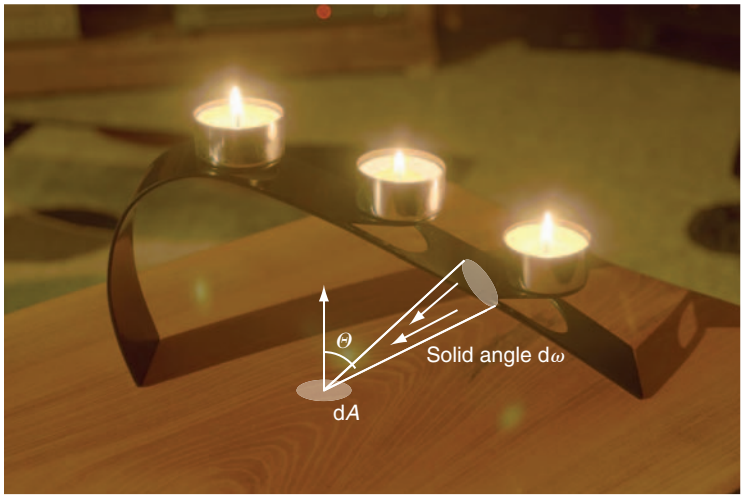


FIGURE 2.4 *Radiance: power incident on a unit surface area* $\mathrm{d}A$ *from a unit set of directions* $\mathrm{d}\omega$.

from the camera sensor to radiance, and provided pixels are neither under nor overexposed [268,269].

Each of the quantities given in Table 2.1 may also be defined per unit wavelength interval and is then referred to as spectral radiance $L_{e,\lambda}$, spectral flux $P_{e,\lambda}$, and so on. The subscript "e" indicates radiometric quantities and differentiates them from photometric quantities that are discussed in the following section. In the remainder of this book, these subscripts are dropped unless this leads to confusion.

Light may be considered to consist of photons that can be emitted, reflected, transmitted, and absorbed. Photons normally travel in straight lines until they hit a surface. The interaction between photons and surfaces is twofold. Photons may be absorbed by the surface where they are converted into thermal energy, or they may be reflected in some direction. The distribution of reflected directions given

FIGURE 2.5 *Bricks, stonework, and rocks are examples of materials that tend to reflect light mostly diffusely. The bright lighting conditions under which this photograph was taken would cause this image to look overall bright, but without a large variation of lighting.*

an angle of incidence gives rise to a surface's appearance. Matte surfaces distribute light almost evenly in all directions (Figure 2.5), whereas glossy and shiny surfaces reflect light in a preferred direction. Mirrors are the opposite of matte surfaces and emit light specularly into almost a single direction. This causes highlights that may be nearly as strong as light sources (Figure 2.6). The depiction of specular surfaces may, therefore, require HDR techniques for accuracy.

For the purpose of lighting simulations, the exact distribution of light reflected off surfaces as function of angle of incidence is important (compare Figures 2.5 and 2.6). It may be modeled with bidirectional reflection distribution functions (BRDFs), which then become part of the surface material description. Advanced

FIGURE 2.6 *The windscreen of this car causes highlights that are nearly as strong as the light source it reflects.*

rendering algorithms use this information to compute how light is distributed in the scene such that an HDR image of the scene may be generated [141,75].

2.2 PHOTOMETRY

Surfaces reflect light, and by doing so, they may alter the spectral composition of it. Thus, reflected light conveys spectral information of both the light source illuminating a surface point and the reflectance of the surface at that point.

There are many wavelengths that are not detectable by the human eye, which is sensitive to wavelengths approximately between 380 and 830 nm. Within this

range, the human eye is not equally sensitive to all wavelengths. In addition, there are fluctuations of sensitivity to the spectral composition of light between individuals. However, these fluctuations are small enough that the spectral sensitivity of any human observer with normal vision may be approximated with a single curve. Such a curve is proposed by the Commission Internationale de l'Eclairage (CIE) and is known as the "$V(\lambda)$ curve" (pronounced *vee-lambda*) or the "CIE photopic luminous efficiency curve." This curve is plotted in Figure 2.7.

Since we are typically interested in how humans perceive light, its spectral composition may be weighted according to $V(\lambda)$. The science of measuring light in units that are weighted in this fashion is called "photometry." All radiometric terms introduced in the previous section have photometric counterparts, which are listed

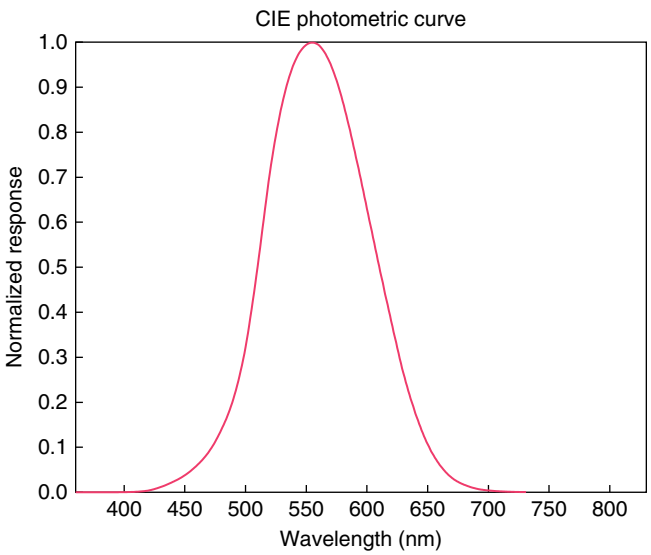


FIGURE 2.7 *CIE standard observer photopic luminous efficiency curve.*

Quantity	Unit
Luminous power (P_v)	lm (lumen)
Luminous energy (Q_v)	lm s
Luminous exitance (M_v)	lm/m^2
Illuminance (E_v)	lm/m^2
Luminous intensity (I_v)	lm/sr = cd (candela)
Luminance (L_v)	cd/m^2 = nit

TABLE 2.2 *Photometric quantities.*

in Table 2.2. By spectrally weighting radiometric quantities with $V(\lambda)$, they are converted into photometric quantities.

Luminous flux or luminous power is photometrically weighted radiant flux. It is measured in lumens, which is defined as $1/683$ W of radiant power at a frequency of 540×10^{12} Hz. This frequency corresponds to the wavelength for which humans are maximally sensitive (around 555 nm). If luminous flux is measured over a differential solid angle, the quantity obtained is luminous intensity and is given in lumen per steradian. One lumen per steradian is equivalent to one candela. Both luminous exitance and illuminance are given in lumens per square meter, whereas luminance is specified in candela per square meter (formerly "nits").

Luminance is a perceived quantity. It is photometrically weighted radiance and constitutes an approximate measure of how bright a surface appears. Luminance is the most relevant photometric unit to HDR imaging. Spectrally weighting radiance amounts to multiplying each spectral component with the corresponding value given by the weight function and then integrating all the results:

$$L_v = \int_{380}^{830} L_{e,\lambda}\, V(\lambda) \mathrm{d}\lambda$$

The consequence of this equation is that there are many different spectral compositions of radiance L_e possible that would cause the same luminance value L_v. It is, therefore, not possible to apply the above formula and expect the resulting luminance value to be a unique representation of the associated radiance value.

The importance of luminance in HDR imaging lies in the fact that it provides a natural boundary of visible wavelengths. Any wavelength outside the visible range does not need to be recorded, stored, or manipulated, since human vision is not capable of detecting those wavelengths. Many tone-reproduction operators first extract a luminance value from the red, green, and blue components of each pixel prior to reducing the dynamic range, since large variations in luminance over orders of magnitude have greater bearing on perception than extremes of color (see also Section 7.6.2).

2.3 COLORIMETRY

The field of colorimetry is concerned with assigning numbers to physically defined stimuli such that stimuli with the same specification look alike, that is, they match. One of the main results from color-matching experiments is that over a wide range of conditions almost all colors may be visually matched by adding light from three suitably pure stimuli. These three fixed stimuli are called "primary stimuli." Color-matching experiments take three light sources and project them to one side of a white screen. A fourth light source, the target color, is projected to the other side of the screen. Participants in the experiments are given control over the intensity of each of the three primary light sources and are asked to match the target color.

For each spectral target, the intensity of the three primaries may be adjusted to create a match. By recording the intensities of the three primaries for each target wavelength, three functions $\bar{r}(\lambda)$, $\bar{g}(\lambda)$, and $\bar{b}(\lambda)$ may be created. These are called "color-matching functions." In Figure 2.8, the color-matching functions obtained by Stiles and Burch are plotted. They used primary light sources that were nearly monochromatic with peaks centered around $\lambda_R = 645.2\,\text{nm}$, $\lambda_G = 525.3\,\text{nm}$, and $\lambda_B = 444.4\,\text{nm}$ [306]. The stimuli presented to the observers in these experiments span 10° of visual angle; hence, these functions are called "10° color-matching functions." Since the recorded responses vary only a small amount among observers,

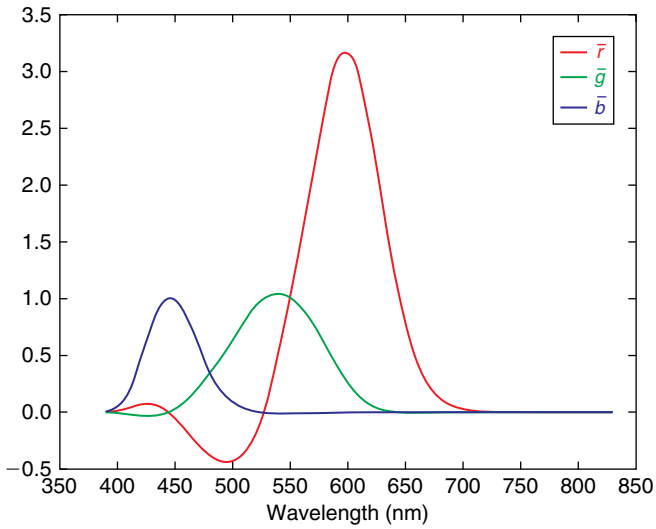


FIGURE 2.8 *Stiles and Burch 1959 10° color-matching functions.*

these color-matching functions are representative of normal human vision. As a result, they were adopted by the CIE to describe the "CIE 1964 Standard Observer."

Thus, a linear combination of three spectral functions yields a fourth Q_λ that may be visually matched to a linear combination of primary stimuli:

$$Q_\lambda = \bar{r}(\lambda)R + \bar{g}(\lambda)G + \bar{b}(\lambda)B$$

Here, R, G, and B are scalar multipliers. Since the primaries are fixed, the stimulus Q_λ may be represented as a triplet by listing R, G, and B. This (R, G, B) triplet is then called the "tristimulus value of Q."

For any three real primaries, it is sometimes necessary to supply a negative amount to reach some colors; that is, there may be one or more negative components of a tristimulus value. Since it is simpler to deal with a color space whose

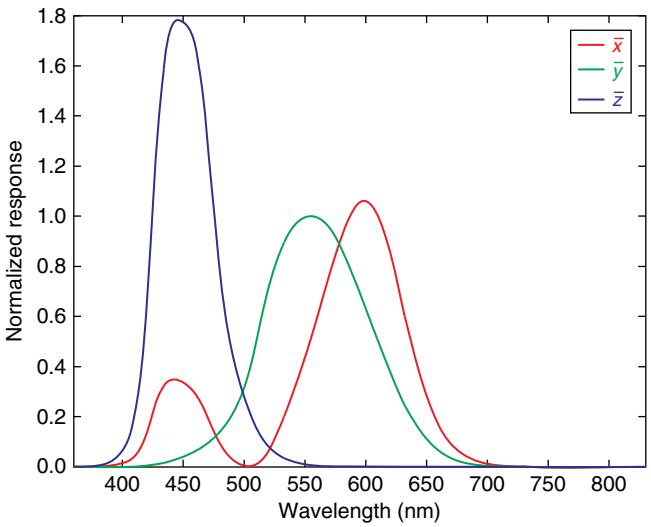


FIGURE 2.9 *CIE 1931 2° XYZ color-matching functions.*

tristimulus values are always positive, the CIE has defined alternative color-matching functions that are chosen such that any color may be matched with positive primary coefficients.[1] These color-matching functions are named $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$, and are plotted in Figure 2.9. These functions are the result of experiments where the stimulus spanned 2° of visual angle and are, therefore, known as the "CIE 1931 Standard Observer" [372].

1 Real or realizable primaries are those that can be obtained by physical devices. For such primaries, it is not possible to supply negative amounts because light cannot be subtracted from a scene. However, although less desirable in practice, there is no mathematical reason why a tristimulus value could not be converted such that it is represented by a different set of primaries. Some of the values may then become negative. Such conversion issues are discussed further in Section 2.4.

A spectral stimulus may now be matched in terms of these color-matching functions:

$$Q_\lambda = \bar{x}(\lambda)\ X + \bar{y}(\lambda)\ Y + \bar{z}(\lambda)\ Z$$

For a given stimulus Q_λ, the tristimulus values (X, Y, Z) are obtained by integration:

$$X = \int_{380}^{830} Q_\lambda\, \bar{x}(\lambda)\, \mathrm{d}\lambda$$

$$Y = \int_{380}^{830} Q_\lambda\, \bar{y}(\lambda)\, \mathrm{d}\lambda$$

$$Z = \int_{380}^{830} Q_\lambda\, \bar{z}(\lambda)\, \mathrm{d}\lambda$$

The CIE XYZ matching functions are defined such that a theoretical equal-energy stimulus, which would have unit radiant power at all wavelengths, maps to tristimulus value $(1,1,1)$. Further, note that $\bar{y}(\lambda)$ is equal to $V(\lambda)$ — another intentional choice by the CIE. Thus, Y represents photometrically weighted quantities.

For any visible colors, the tristimulus values in XYZ space are all positive. However, as a result, the CIE primaries are not realizable by any physical device. Such primaries are called "imaginary" as opposed to realizable primaries, which are called "real."[2]

Associated with tristimulus values are chromaticity coordinates, which may be computed from tristimulus values:

$$x = \frac{X}{X + Y + Z}$$

2 This has nothing to do with the mathematical formulation of "real" and "imaginary" numbers.

$$y = \frac{Y}{X+Y+Z}$$

$$z = \frac{Z}{X+Y+Z} = 1 - x - y$$

Since z is known if x and y are known, only the latter two chromaticity coordinates need to be kept. Chromaticity coordinates are relative, which means that within a given system of primary stimuli, two colors with the same relative spectral power distribution will map to the same chromaticity coordinates. An equal-energy stimulus will map to coordinates $(x = 1/3, y = 1/3)$.

Chromaticity coordinates may be plotted in a chromaticity diagram with two axes. A CIE xy chromaticity diagram is shown in Figure 2.10. All monochromatic wavelengths map to a position along the curved boundary, which is called the "spectral locus" and resembles a horseshoe shape. The line between red and blue is called

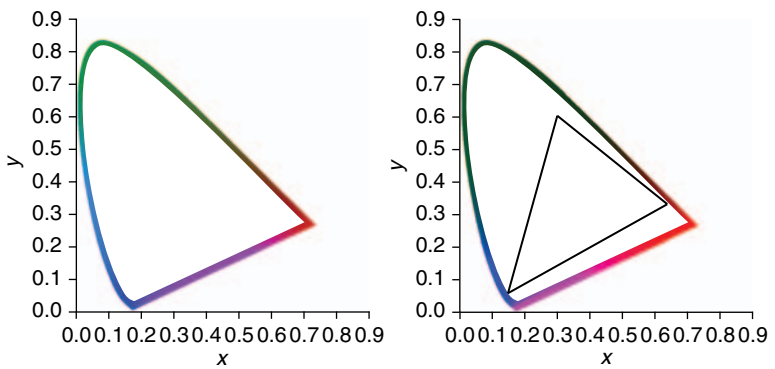


FIGURE 2.10 *CIE xy chromaticity diagram showing the range of colors that humans can distinguish (left). On the right, the triangular gamut spanned by the primaries defined by the ITU-R Rec BT.709 color space [138] is shown.*

the "purple line" and represents the locus of additive mixtures of short- and long-wave stimuli.

The three primaries used for any given color space will map to three points in a chromaticity diagram and thus span a triangle. This triangle contains the range of colors that may be represented by these primaries (assuming nonnegative tristimulus values and a linear color space). The range of realizable colors given a set of primaries is called the "color gamut." Colors that are not representable in a given color space are called "out-of-gamut."

The gamut for the primaries defined by ITU-R Recommendation BT.709 is shown on the right of Figure 2.10. These primaries are a reasonable approximation to most CRT computer monitors and officially define the boundaries of the sRGB color space [307] (see Section 2.12). The triangular region shown in this figure marks the range of colors that may be displayed on a standard monitor. The colors outside this triangle cannot be represented on most displays. They also cannot be stored in an sRGB file, such as the one used for this figure. We are, therefore, forced to show incorrect colors outside the sRGB gamut in all the chromaticity diagrams in this book.

The diagrams in Figure 2.10 only show two dimensions of what is a three-dimensional space. The third dimension, luminance, goes out of the page, and the color gamut is really a volume of which a slice is depicted. In the case of the sRGB color space, the gamut is shaped as a six-sided polyhedron, often referred to as the "RGB color cube." This is misleading since the sides are only equal in the encoding (0–255 thrice) and not very equal perceptually.

It may be possible for two stimuli with different spectral radiant power distributions to match against the same linear combination of primaries so that they are represented by the same set of tristimulus values. This phenomenon is called "metamerism." Metameric stimuli map to the same location in a chromaticity diagram, whereas stimuli that appear different map to different locations. The magnitude of the perceived difference between two stimuli may be expressed as the Cartesian distance between the two points in a chromaticity diagram. However, in the 1931 CIE primary system, the chromaticity diagram is not uniform, that is, the distance between two points located in one part of the diagram corresponds to a different perceived color difference than two points located elsewhere in the diagram. Although CIE XYZ is still the basis for all color theory, this nonuniformity has given rise to alternative color spaces that are discussed in the following sections.

2.4 COLOR SPACES

Color spaces encompass two different concepts. First, it is a set of formulas that define a relationship between a color vector or triplet and the standard CIE XYZ color space. This is most often given in the form of a 3×3 color transformation matrix, though there are additional formulas if the space is nonlinear. Second, a color space is a two-dimensional boundary on the volume defined by this vector, usually determined by the minimum value and the maximum value of each primary — the color gamut. Optionally, the color space may have an associated quantization if it has an explicit binary representation. In this section, linear transformations are discussed, whereas subsequent sections introduce nonlinear encodings and quantization.

We can convert from one tristimulus color space to any other tristimulus space using a 3×3 matrix transformation. Usually, the primaries are known by their xy chromaticity coordinates. In addition, the white point needs to be specified, which is given as an xy chromaticity pair (x_W, y_W), plus its maximum luminance Y_W. The white point is the color associated with equal contributions of each primary and is discussed further in Section 2.5.

Given the chromaticity coordinates of the primaries, first, the z chromaticity coordinate for each primary is computed to yield chromaticity triplets for each primary, namely (x_R, y_R, z_R), (x_G, y_G, z_G), and (x_B, y_B, z_B). From the white point's chromaticities and its maximum luminance, the tristimulus values (X_W, Y_W, Z_W) are then calculated. Then, the following set of linear equations is solved for S_R, S_G, and S_B:

$$\begin{aligned} X_W &= x_R\, S_R + x_G\, S_G + x_B\, S_B \\ Y_W &= y_R\, S_R + y_G\, S_G + y_B\, S_B \\ Z_W &= z_R\, S_R + z_G\, S_G + z_B\, S_B \end{aligned}$$

The conversion matrix to convert from RGB to XYZ is then given by the following:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x_R\, S_R & x_G\, S_G & x_B\, S_B \\ y_R\, S_R & y_G\, S_G & y_B\, S_B \\ z_R\, S_R & z_G\, S_G & z_B\, S_B \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

The conversion from XYZ to RGB may be computed by inverting the above matrix.

If the primaries or the white point is unknown, a second-best solution is to use a standard matrix, such as the one specified by the International Telecommunication Union as ITU-R Recommendation BT.709 [138]:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.2405 & -1.5371 & -0.4985 \\ -0.9693 & 1.8760 & 0.0416 \\ 0.0556 & -0.2040 & 1.0572 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

The primaries and white point used to create this conversion matrix are listed in Table 2.3.

There are several standard color spaces; each is used in different fields of science and engineering. They are each reached by constructing a conversion matrix, the above matrix being an example. Several of these color spaces include a nonlinear transform akin to gamma correction, which is explained in Section 2.10. We therefore defer a discussion of other standard color spaces until Section 2.12.

In addition to standard color spaces, most cameras, scanners, monitors, and TVs use their own primaries (called "spectral responsivities" in the case of capturing devices). Thus, each device may use a different color space (see also Section 2.12). Conversion between these color spaces is thus essential for the faithful reproduction of an image on any given display device.

	R	G	B	White
x	0.6400	0.3000	0.1500	0.3127
y	0.3300	0.6000	0.0600	0.3290

TABLE 2.3 *Primaries and white point specified by ITU-R Recommendation BT.709.*

If a color is specified in a device-dependent RGB color space, its luminance may be computed since the Y component in XYZ color space represents luminance L (recall that $V(\lambda)$ equals $\bar{y}(\lambda)$). Thus, a representation of luminance is obtained by computing a linear combination of the red, green, and blue components according to the middle row of the RGB to XYZ conversion matrix. For instance, luminance may be computed from ITU-R Recommendation BT.709 RGB as follows:

$$L = Y = 0.2126\,R + 0.7152\,G + 0.0722\,B \tag{2.1}$$

Finally, an important consequence of color metamerism is that if the spectral responsivities (primaries) associated with a camera are known, as well as the emissive spectra of the three phosphors of a CRT display, we may be able to specify a transformation between the tristimulus values captured with the camera and the tristimulus values of the display, and thus reproduce the captured image on the display. This would, of course, only be possible if the camera and display technologies did not impose restrictions on the dynamic range of captured and displayed data.

2.5 WHITE POINT AND ILLUMINANTS

For the conversion of tristimulus values between a XYZ and a specific RGB color space, the primaries of the RGB color space must be specified. In addition, the white point needs to be known. For a display device, the white point is the color emitted if all three color channels are contributing equally.

Similarly, within a given scene, the dominant light source produces a color cast that affects the appearance of the objects in the scene. The color of a light source (illuminant) may be determined by measuring a diffusely reflecting white patch. The color of the illuminant, therefore, determines the color of a scene that the human visual system normally associates with white.

An often used reference light source is CIE illuminant D_{65}. This light source may be chosen if no further information is available regarding the white point of a device or the illuminant of a scene. Its spectral power distribution is shown in Figure 2.11,

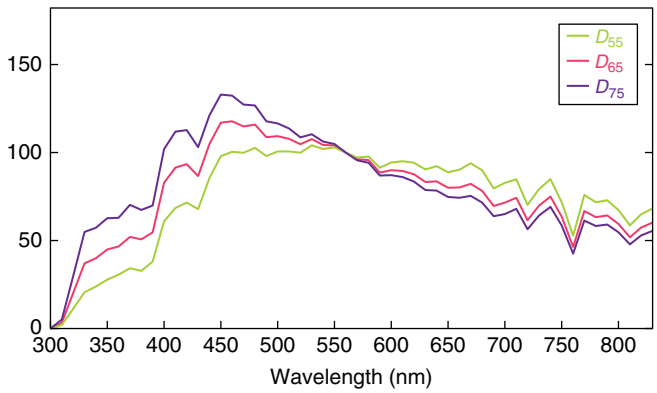


FIGURE 2.11 *Spectral power distribution of CIE illuminants* D_{55}, D_{65}, *and* D_{75}.

along with two related standard illuminants, D_{55} (commonly used in photography) and D_{75}.

Cameras often operate under the assumption that the scene is lit by a specific light source, such as a D_{65}. If the lighting in a scene has a substantially different color, an adjustment to the gain of the red, green, and blue sensors in the camera may be made. This is known as "white balancing" [256]. If the white balance chosen for a particular scene was incorrect, white balancing may be attempted as an image-processing step.

The difference between illuminants may be expressed in terms of chromaticity coordinates, but a more commonly used measure is "correlated color temperature." Consider a blackbody radiator, a cavity in a block of material which is heated to a certain temperature. The spectral power distribution emitted by the walls of this cavity is a function of the temperature of the material only. The color of a blackbody radiator may thus be characterized by its temperature, which is measured in degrees Kelvin (K).

The term "color temperature" refers to color associated with the temperature of a blackbody radiator. The lower the temperature, the redder the appearance of the radiator. For instance, tungsten illumination (approximately 3200 K) appears somewhat yellow. Higher color temperatures have a more bluish appearance. The colors that a blackbody radiator can attain are located on a curve in a chromaticity diagram that is known as the *Planckian locus*.

The term "correlated color temperature" is more generally used for nonblackbody illuminants. It refers to the blackbody's temperature that most closely resembles the perceived color of the given selective radiator under the same brightness and specified viewing conditions. Correlated color temperatures are only applicable to radiators with chromaticity coordinates that are within $0.05u'v'$ units of the Planckian locus [43]. Table 2.4 lists the correlated color temperature of several

Scene	T (in °K)	x	y
Candle flame	1850	0.543	0.410
Sunrise/sunset	2000	0.527	0.413
Tungsten (TV/film)	3200	0.427	0.398
Summer sunlight at noon	5400	0.326	0.343
CIE A (incandescent)	2854	0.448	0.408
CIE B (direct sunlight)	4874	0.384	0.352
CIE C (indirect sunlight)	6774	0.310	0.316
CIE D50 (noon skylight)	5000	0.346	0.359
CIE D65 (average daylight)	6504	0.313	0.329
CIE E (normalized reference)	5500	0.333	0.333
CIE F2 (office fluorescent)	4150	0.372	0.375

TABLE 2.4 *Correlated color temperature T and chromaticity coordinates $(x\,y)$ for common scene types and a selection of CIE luminaires.*

common scene types and CIE luminaires, as well as their associate chromaticity coordinates.

The CIE standard illuminant D_{65} shown in Figure 2.11 is defined as natural daylight with a correlated color temperature of 6504 K. The D_{55} and D_{75} illuminants have correlated color temperatures of 5503 and 7504 K, respectively. Many color spaces are defined with a D_{65} white point. In photography, D_{55} is often used.

Humans are capable of adapting to the chromaticity of the dominant light source in a scene, in particular to white light illuminants of various color temperatures. The impression of color given by a surface depends on its reflectance, as well as the light source illuminating it. If the light source is gradually changed in color, humans will adapt and still perceive the color of the surface the same although light measurements of the surface would indicate a different spectral composition and CIE XYZ tristimulus value [308]. This phenomenon is called "chromatic adaptation." The ability to perceive the color of a surface independent of the light source illuminating it is called "color constancy."

Typically, when viewing a real scene, an observer would be chromatically adapted to that scene. When an image of the same scene is displayed on a display device, the observer would be adapted to the display device and the scene in which the observer views the image. It is reasonable to assume that these two states of adaptation will generally be different. As such, the image shown is likely to be perceived differently from the real scene. Accounting for such differences should be an important aspect of HDR imaging and in particular tone-reproduction. Unfortunately, too many tone-reproduction operators ignore these issues, although the Multiscale Observer Model, iCAM, and the photoreceptor-based operator include a model of chromatic adaptation (see Sections 8.2.1, 8.2.2, and 8.1.1), and Akyuz et al. have shown that tone reproduction and color appearance modeling may be separated into two separate steps [10].

In 1902, von Kries speculated that chromatic adaptation is mediated by the three cone types in the retina [167]. Chromatic adaptation occurs as the red, green, and blue cones each independently adapt to the illuminant.

A model of chromatic adaptation may thus be implemented by transforming tristimulus values into a cone response domain and then individually scaling the red, green, and blue components according to the current and desired illuminants.

There exist different definitions of cone response domains, leading to different transforms. The first cone response domain is given by the LMS color space, with L, M, and S standing for long, medium, and short wavelengths. The matrix that converts between XYZ and LMS lies at the heart of the von Kries transform and is denoted $M_{\text{von Kries}}$:

$$M_{\text{von Kries}} = \begin{bmatrix} 0.3897 & 0.6890 & -0.0787 \\ -0.2298 & 1.1834 & 0.0464 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$$

$$M^{-1}_{\text{von Kries}} = \begin{bmatrix} 1.9102 & -1.1121 & 0.2019 \\ 0.3710 & 0.6291 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$$

As the LMS cone space represents the response of the cones in the human visual system, it is a useful starting place for computational models of human vision. It is also a component in the iCAM color appearance model (see Section 8.2.2). The relative response as function of wavelength is plotted in Figure 2.12.

A newer cone response domain is given by the Bradford chromatic adaptation transform [148,190] (see Figure 2.13):

$$M_{\text{Bradford}} = \begin{bmatrix} 0.8951 & 0.2664 & -0.1614 \\ -0.7502 & 1.7135 & 0.0367 \\ 0.0389 & -0.0685 & 1.0296 \end{bmatrix}$$

$$M^{-1}_{\text{Bradford}} = \begin{bmatrix} 0.9870 & -0.1471 & 0.1600 \\ 0.4323 & 0.5184 & 0.0493 \\ -0.0085 & 0.0400 & 0.9685 \end{bmatrix}$$

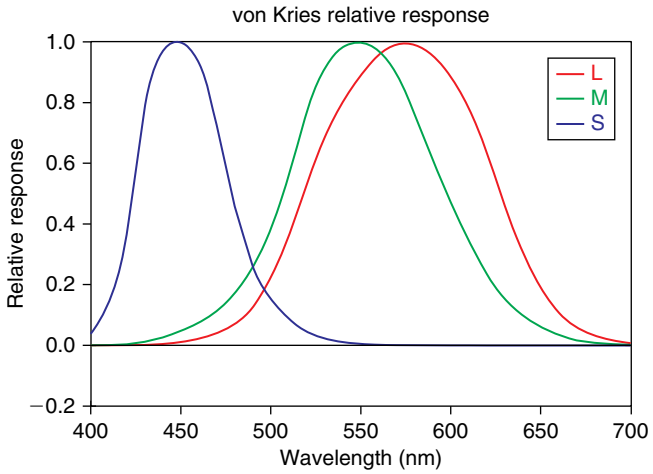


FIGURE 2.12 *Relative response functions for the von Kries chromatic adaptation transform. (Reprinted from [94]. Image courtesy of Society for Imaging Science and Technology Publications.)*

A third chromatic adaptation transform is used in the CIECAM02 color appearance model, which is described in Section 2.9 (see Figure 2.14):

$$M_{\mathrm{CAT02}} = \begin{bmatrix} 0.7328 & 0.4296 & -0.1624 \\ -0.7036 & 1.6975 & 0.0061 \\ 0.0030 & 0.0136 & 0.9834 \end{bmatrix}$$

$$M_{\mathrm{CAT02}}^{-1} = \begin{bmatrix} 1.0961 & -0.2789 & 0.1827 \\ 0.4544 & 0.4735 & 0.0721 \\ -0.0096 & -0.0057 & 1.0153 \end{bmatrix}$$

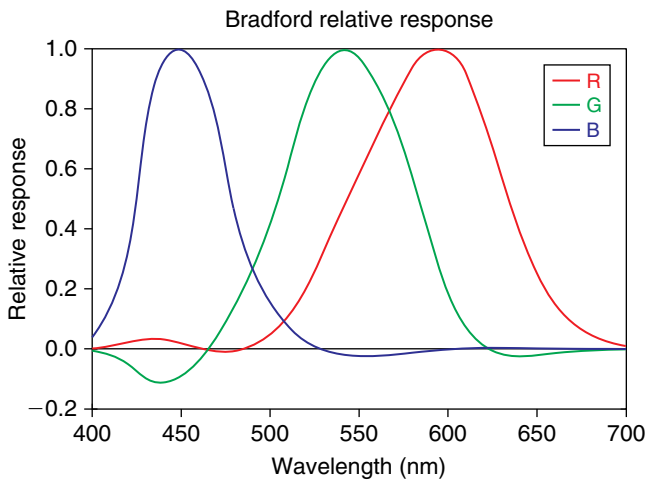


FIGURE 2.13 *Relative response functions for the Bradford chromatic adaptation transform. (Reprinted from [94]. Image courtesy of Society for Imaging Science and Technology Publications.)*

The above three chromatic adaptation transforms may be used to construct a matrix that transforms XYZ tristimulus values for a given white point to a new white point [313]. If the source white point is given as (X_S, Y_S, Z_S) and the destination white point as (X_D, Y_D, Z_D), their transformed values are

$$\begin{bmatrix} \rho_S \\ \gamma_S \\ \beta_S \end{bmatrix} = M_{cat} \begin{bmatrix} X_S \\ Y_S \\ Z_S \end{bmatrix}$$

$$\begin{bmatrix} \rho_D \\ \gamma_D \\ \beta_D \end{bmatrix} = M_{cat} \begin{bmatrix} X_D \\ Y_D \\ Z_D \end{bmatrix}$$

where M_{cat} is one of the three chromatic adaptation matrices $M_{\text{von Kries}}$, M_{Bradford}, or M_{CAT02}. A chromatic adaptation matrix for these specific white points may be constructed by concatenating the above von Kries or Bradford matrices with a diagonal matrix that independently scales the three cone responses:

$$M = M_{\text{cat}}^{-1} \begin{bmatrix} \rho_{\text{D}}/\rho_{\text{S}} & 0 & 0 \\ 0 & \gamma_{\text{D}}/\gamma_{\text{S}} & 0 \\ 0 & 0 & \beta_{\text{D}}/\beta_{\text{S}} \end{bmatrix} M_{\text{cat}}$$

Chromatically adapting an XYZ tristimulus value is now a matter of transforming it with matrix M:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = M \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

where (X', Y', Z') is the CIE tristimulus value whose appearance under the target illuminant most closely matches the original XYZ tristimulus under the source illuminant.

Chromatic adaptation transforms are useful for preparing an image for display under different lighting conditions. Thus, if the scene was lit by daylight, and an image of that scene is viewed under tungsten lighting, a chromatic adaptation transform may be used to account for this difference. After applying the chromatic adaptation transform, the (X', Y', Z') tristimulus values then need to be converted to an RGB color space with a matrix that takes into account the white point of the display environment. Thus, if the image is viewed under tungsten lighting, the XYZ-to-RGB transformation matrix should be constructed using the white point of a tungsten light source.

As an example, Figure 2.15 shows an image lit with daylight approximating D_{65}.[3] This figure shows the image prepared for several different viewing environments. In each case, the CAT02 chromatic adaptation transform was used, and the

3 This image was taken in a conservatory in Rochester, NY, under cloud cover. The CIE D_{65} standard light source was derived from measurements originally made of similar daylight conditions in Rochester.

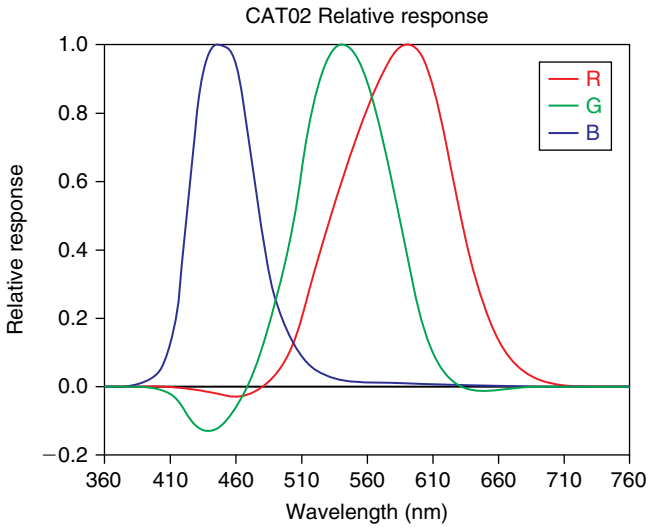


FIGURE 2.14 *Relative response functions for the CAT02 chromatic adaptation transform. (Reprinted from slides by Mark Fairchild.)*

conversion to RGB color space was achieved by constructing a conversion matrix with the appropriate white point.

The difference between the three different chromatic adaptation transforms is illustrated in Figure 2.16. Figure 2.16 also shows a chromatic adaptation performed directly in XYZ space, here termed "XYZ scaling." The scene depicted here was created with only outdoors lighting available and was taken in the same conservatory as the images in Figure 2.15. Thus, the lighting in this scene would be reasonably well approximated with a D_{65} luminant. In Figure 2.16, transforms from D_{65} to F2 are given.

FIGURE 2.15 *CAT02 chromatic adaptation. In reading order: original image, followed by five images chromatically adapted from* D_{65} *to incandescent, tungsten,* D_{50}, E, *and* F2.

FIGURE 2.16 Comparison between different chromatic adaptation transforms. In reading order: original image, followed by von Kries, Bradford, and CAT02 transforms. The final image is the chromatic adaptation transform applied directly in XYZ space. The transform is from D_{50} to D_{65}.

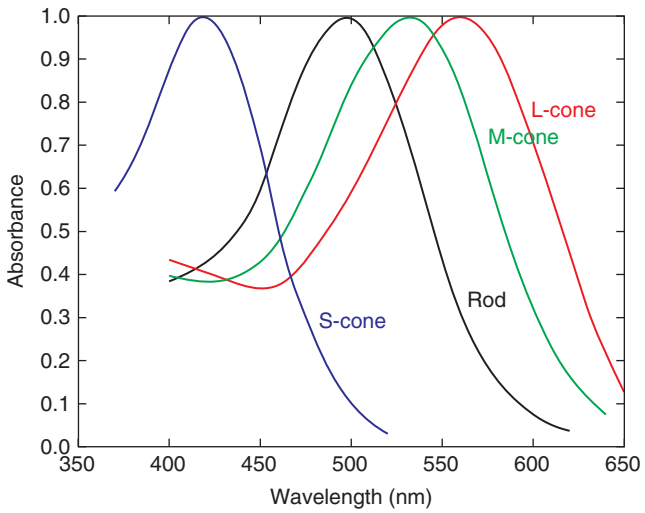


FIGURE 2.17 *Spectral absorbance spectra for the L, M, and S cones, as well as the rods (after [53]).*

The spectral sensitivities of the cones in the human visual system are broadband, that is, each of the red, green, and blue cone types (as well as the rods) is sensitive to a wide range of wavelengths, as indicated by their absorbance spectra shown in Figure 2.17 [53]. As a result, there is significant overlap between the different cone types, although their peak sensitivities lie at different wavelengths.

2.6 SPECTRAL SHARPENING

It is possible to construct new spectral response functions that are more narrowband by computing a linear combination of the original response functions. The graphs

of the resulting response functions look sharper, and the method is therefore called "spectral sharpening." Within a chromaticity diagram, the three corners of the color gamut lie closer to the spectral locus, or even outside, and therefore, the gamut is "wider" so that a greater range of visible colors can be represented.

A second advantage of applying such a transform is that the resulting tristimulus values become more decorrelated. This has advantages in color constancy algorithms, or algorithms that aim to recover surface reflectance from an image which has recorded the combined effect of surface reflectance and illuminance [19].

It also helps reduce visible errors in color-rendering algorithms [341]. Renderings made in the sRGB color space and images rendered in a sharpened color space can be compared with full spectral renderings. Here, it turns out that the spectrally sharpened images are closer to the full spectral renderings than their sRGB equivalents.

2.7 COLOR OPPONENT SPACES

With a 3×3 matrix, pixel data may be rotated into different variants of RGB color spaces to account for different primaries. A feature shared by all RGB color spaces is that for natural images, correlations exist between the values in each color channel. In other words, if a pixel of a natural image has a large value for the red component, the probability of also finding a large value for the green and blue components is high. Thus, the three channels are highly correlated.

An example image is shown in Figure 2.18. A set of randomly selected pixels is plotted three times in the same figure, where the axes of the plot are R-G, R-B, and G-B, respectively. This plot shows a point cloud of pixel data at an angle of around 45°, and no matter which channel is plotted against which. Thus, for this natural image, strong correlations exist between the channels in RGB color space.

This means that the amount of information carried by the three values comprising a pixel is less than three times the amount of information carried by each one of the values. Thus, each color pixel carries some unquantified amount of redundant information.

The human visual system deals with a similar problem. The information captured by the photoreceptors needs to be transmitted to the brain through the optic

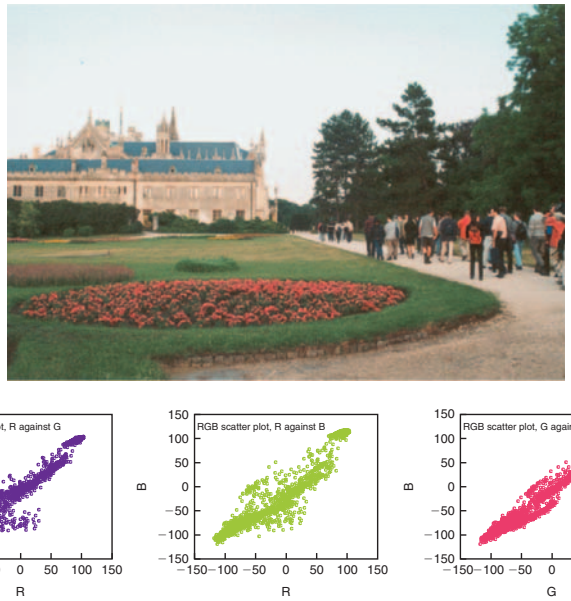


FIGURE 2.18 *Scatter plots of RGB pixels randomly selected from the image at the top.*

nerve. The amount of information that can pass through the optic nerve is limited and constitutes a bottleneck. In particular, the number of photoreceptors in the retina is far larger than the number of nerve endings that connect the eye to the brain.

After light is absorbed by the photoreceptors, a significant amount of processing occurs in the next several layers of cells before the signal leaves the eye. One type of processing is a color space transformation to a *color opponent space*. Such a color space is characterized by three channels, namely a luminance channel, a red–green channel, and a yellow–blue channel.

The luminance channel ranges from dark to light and bears resemblance to the Y channel in CIE XYZ color space. The red–green channel ranges from red to green via neutral gray. The yellow–blue channel encodes the amount of blue versus the amount of yellow in a similar way to the red–green channel (Figure 2.19). This

FIGURE 2.19 *Original image (top left) split into a luminance channel (top right), a yellow–blue channel (bottom left) and a red–green channel (bottom right). For the purpose of visualization, the images depicting the yellow–blue, and red–green channels are shown with the luminance component present.*

encoding of chromatic information is the reason that humans are able to describe colors as reddish yellow (orange) or greenish blue. However, colors such as bluish yellow and reddish green are never described because of this encoding (see [243], p. 109).

It is possible to analyze sets of natural images by means of principal components analysis (PCA) [280]. This technique rotates multidimensional data such that the axis align with the data as well as possible. Thus, the most important axis aligns with the direction in space that shows the largest variation of data points. This is the first principal component. The second principal component describes the direction accounting for the second most variation in the data. This rotation therefore decorrelates the data.

If the technique is applied to images encoded in LMS color space,[4] that is, images represented in a format as thought to be output by the photoreceptors, a new set of decorrelated axes is produced. The surprising result is that the application of PCA to a set of natural images produces a color space that is closely matched to the color opponent space the human visual system employs [280].

A scatter plot of the image of Figure 2.18 in a color opponent space ($L\alpha\beta$, discussed later in this section) is given in Figure 2.20. Here, the point clouds are reasonably well aligned with one of the axes, indicating the data is now decorrelated. The elongated shape of the point clouds indicates the ordering of the principal axes, luminance being most important and therefore most elongated.

The decorrelation of data may be important for instance for color correction algorithms. What would otherwise be a complicated three-dimensional problem may be cast into three simpler one-dimensional problems by solving the problem in a color opponent space [271].

At the same time, the first principal component, the luminance channel, accounts for the greatest amount of variation, whereas the two chromatic color opponent channels carry less information. Converting an image into a color space with a luminance channel and two chromatic channels thus presents an opportunity to compress data since the latter channels would not require the same number of bits as the luminance channel to accurately represent the image.

4 LMS describes three color axes that have peaks at "long," "medium," and "short" wavelengths. See also Section 2.7.3.

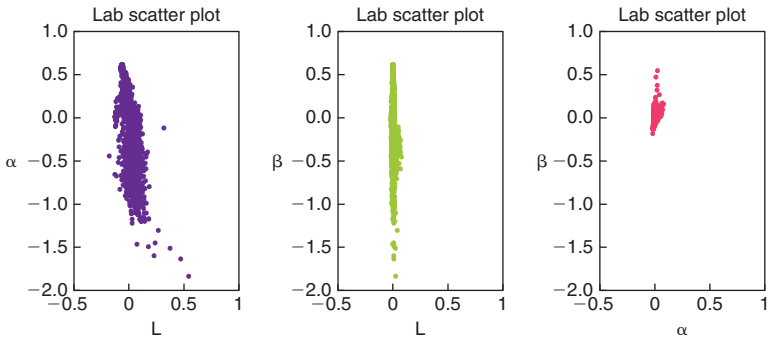


FIGURE 2.20 *Scatter plots of $L\alpha\beta$ pixels randomly selected from the image of Figure 2.18.*

The color opponent space $L\alpha\beta$ that results from applying PCA to natural images may be approximated by the following matrix transform, which converts between $L\alpha\beta$ and LMS (see Section 2.5):

$$\begin{bmatrix} L \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix}$$

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -2 & 0 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3}}{3} & 0 & 0 \\ 0 & \frac{\sqrt{6}}{6} & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} L \\ \alpha \\ \beta \end{bmatrix}$$

This color space has proved useful in algorithms such as the transfer of color between images, where the colors are borrowed from one image and applied to a second image [271]. This algorithm computes means and standard deviations for each channel separately in both source and target images. Then, the pixel data in the target image are shifted and scaled such that the same mean and standard deviation as the source image are obtained. Applications of color transfer include the work of colorists, compositing, and matching rendered imagery with live video footage in mixed reality applications.

In addition, human sensitivity to chromatic variations is lower than that to changes in luminance. Chromatic channels may therefore be represented at a lower spatial resolution than the luminance channel. This feature may be exploited in image encodings by sampling the image at a lower resolution for the color opponent channels than for the luminance channel. This is demonstrated in Figure 2.21, where the full resolution image is shown on the left. The spatial resolution of the red–green and yellow–blue channels is reduced by a factor of two for each subsequent image. In Figure 2.22, the luminance channel was also reduced by a factor of two. The artifacts in the top row are much more benign than the ones in the bottom row.

Subsampling of chromatic channels is used for instance in the $\mathbf{YC_BC_R}$ encoding, which is part of the JPEG file format and is also part of various broadcast standards, including HDTV [256]. Conversion from RGB to $\mathbf{YC_BC_R}$ and back as used for JPEG is given by

$$\begin{bmatrix} Y \\ C_\mathrm{B} \\ C_\mathrm{R} \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.168 & -0.333 & 0.498 \\ 0.498 & -0.417 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.000 & 1.397 \\ 1.000 & -0.343 & -0.711 \\ 1.000 & 1.765 & 0.000 \end{bmatrix} \begin{bmatrix} Y \\ C_\mathrm{B} \\ C_\mathrm{R} \end{bmatrix}$$

This conversion is based on ITU Recommendation BT.601 [256].

Commonly used opponent color spaces are the CIELUV, CIELAB, and IPT color spaces, which are discussed next.

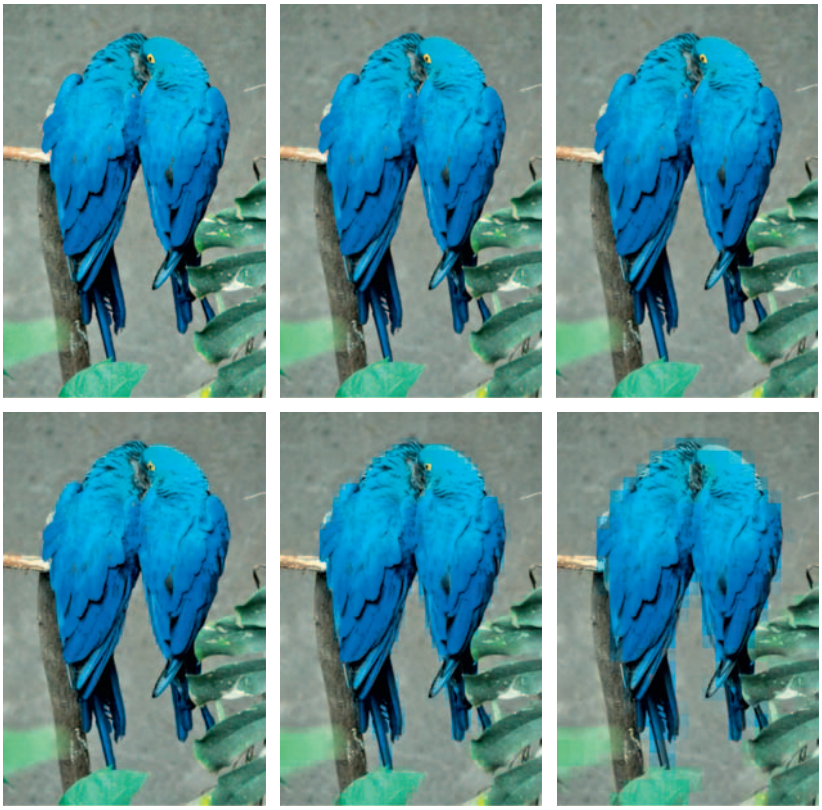


FIGURE 2.21 The red–green and yellow–blue channels are reduced in spatial resolution by a factor of 1, 2, 4, 8, 16, and 32.

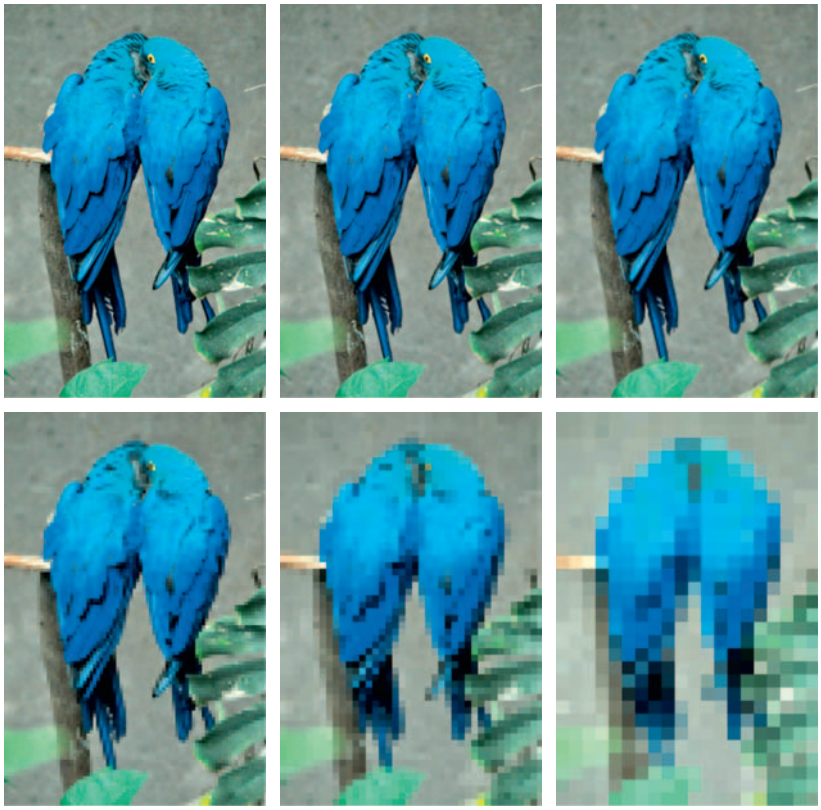


FIGURE 2.22 *All three channels in $L\alpha\beta$ space are reduced in spatial resolution by a factor of 1, 2, 4, 8, 16, and 32.*

2.7.1 CIELUV

The aim of both CIE 1976 $L^*u^*v^*$ and CIE 1976 $L^*a^*b^*$ (abbreviated to CIELUV and CIELAB) is to provide perceptually uniform color spaces. This means that Euclidian distances in such spaces can be used to assess perceived color differences. Both CIELUV and CIELAB are perceptually uniform to a good degree, but based on the newer psychophysical data, it has become apparent that some nonuniformities remain. For this reason, the more recent IPT color space was developed, which is discussed in Section 2.7.3.

For both the CIELUV and CIELAB color spaces, it is assumed that a stimulus (X, Y, Z) is formed by a white reflecting surface that is lit by a known illuminant with tristimulus values $(X_\mathrm{n}, Y_\mathrm{n}, Z_\mathrm{n})$. The conversion from CIE 1931 tristimulus values to CIELUV is then given by

$$L^* = 116\left(\frac{Y}{Y_\mathrm{n}}\right)^{1/3} - 16$$
$$u^* = 13\,L^*(u' - u'_\mathrm{n})$$
$$v^* = 13\,L^*(v' - v'_\mathrm{n})$$

The above conversion is under the constraint that $Y/Y_\mathrm{n} > 0.008856$. For ratios smaller than 0.008856, L^*_m is applied:

$$L^*_\mathrm{m} = 903.3\,\frac{Y}{Y_\mathrm{n}}$$

The primed quantities in the above equations are computed from (X, Y, Z) as follows:

$$u' = \frac{4\,X}{X + 15\,Y + 3\,Z} \qquad u'_\mathrm{n} = \frac{4\,X_\mathrm{n}}{X_\mathrm{n} + 15\,Y_\mathrm{n} + 3\,Z_\mathrm{n}}$$
$$v' = \frac{9\,Y}{X + 15\,Y + 3\,Z} \qquad v'_\mathrm{n} = \frac{9\,Y_\mathrm{n}}{X_\mathrm{n} + 15\,Y_\mathrm{n} + 3\,Z_\mathrm{n}}$$

This transformation creates a more or less uniform color space such that equal distances anywhere within this space encode equal perceived color differences. It is

therefore possible to measure the difference between two stimuli (L_1^*, u_1^*, v_1^*) and (L_2^*, u_2^*, v_2^*) by encoding them in CIELUV space and applying the following color difference formula:

$$\Delta E_{uv}^* = \left[(\Delta L^*)^2 + (\Delta u^*)^2 + (\Delta v^*)^2 \right]^{1/2}$$

where $\Delta L^* = L_1^* - L_2^*$, etc.

In addition, u' and v' may be plotted on separate axes to form a chromaticity diagram, as shown in Figure 2.23. Equal distances in this diagram represent approximately equal perceptual differences. For this reason, in the remainder of this book CIE (u', v'), chromaticity diagrams are shown rather than perceptually nonuniform CIE (x, y) chromaticity diagrams.

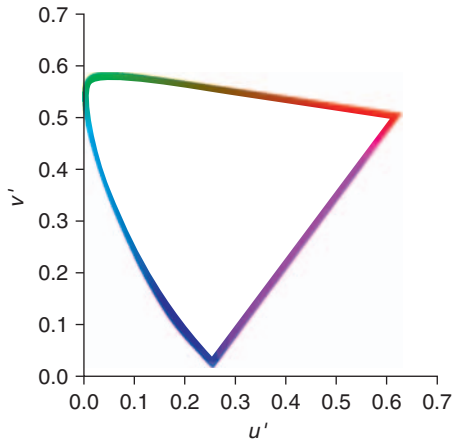


FIGURE 2.23 CIE (u', v') chromaticity diagram showing the range of colors that humans can distinguish.

2.7.2 CIELAB

The CIELAB color space follows an approach similar to the one taken by CIELUV. For the ratios X/X_n, Y/Y_n, and Z/Z_n each larger than 0.008856, the color space is defined by

$$L^* = 116\left(\frac{Y}{Y_n}\right)^{1/3} - 16$$
$$a^* = 500\left[\left(\frac{X}{X_n}\right)^{1/3} - \left(\frac{Y}{Y_n}\right)^{1/3}\right]$$
$$b^* = 200\left[\left(\frac{Y}{Y_n}\right)^{1/3} - \left(\frac{Z}{Z_n}\right)^{1/3}\right]$$

For ratios smaller than 0.008856, the modified quantities L^*_m, a^*_m, and b^*_m may be computed as follows:

$$L^*_m = 903.3\,\frac{Y}{Y_n} \qquad \text{for } \frac{Y}{Y_n} \le 0.008856$$
$$a^*_m = 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right]$$
$$b^*_m = 500\left[f\left(\frac{Y}{Y_n}\right) - f\left(d\,\frac{Z}{Z_n}\right)\right]$$

The function $f(.)$ takes a ratio as argument in the above equations. If either of these ratios is denoted as r, then $f(r)$ is defined as

$$f(r) = \begin{cases} (r)^{1/3} & \text{for } r > 0.008856 \\ 7.787\,r + \dfrac{16}{116} & \text{for } r \le 0.008856 \end{cases}$$

Within this color space, which is also approximately perceptually linear, the difference between two stimuli may be quantified with the following color difference formula:

$$\Delta E^*_{ab} = \left[(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2\right]^{1/2}$$

The reason for the existence of both of these color spaces is largely historical. Both color spaces are in use today, with CIELUV more common in the television and display industries and CIELAB in the printing and materials industries [308].

2.7.3 IPT

The IPT color space is designed to be perceptually uniform, improving upon CIELAB in particular with respect to hue uniformity[5] [77]. It contains a channel I encoding intensity[6], a channel P (standing for *protan*) that roughly corresponds to red–green color opponency, and a blue–yellow channel T (for *tritan*). The space is defined with respect to CIE XYZ and begins by a linear transform to LMS cone excitation space (see also p. 39):

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.4002 & 0.7075 & -0.0807 \\ -0.2280 & 1.1500 & 0.0612 \\ 0.0000 & 0.0000 & 0.9184 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

This matrix defines LMS cone responses under the assumption of D65 illumination. These responses are then nonlinearly transformed:

$$L' = \begin{cases} L^{0.43} & \text{if } L \geq 0, \\ -(-L)^{0.43} & \text{if } L < 0; \end{cases}$$

$$M' = \begin{cases} M^{0.43} & \text{if } M \geq 0, \\ -(-M)^{0.43} & \text{if } M < 0; \end{cases}$$

$$S' = \begin{cases} S^{0.43} & \text{if } S \geq 0, \\ -(-S)^{0.43} & \text{if } S < 0 \end{cases}$$

5 Hue, as defined on p. 64, is the attribute of color perception denoted by red, green, blue, yellow, purple, etc.

6 "Intensity" is used here colloquially. The computation of I involves a compressive step that suggests that this channel correlates with brightness, akin to the cube root in the computation of the L channel in CIELAB and CIELUV.

Finally, the IPT color space is computed by means of a second linear transformation:

$$\begin{bmatrix} I \\ P \\ T \end{bmatrix} = \begin{bmatrix} 0.4000 & 0.4000 & 0.2000 \\ 4.4550 & -4.8510 & 0.3960 \\ 0.8056 & 0.3572 & -1.1628 \end{bmatrix} \begin{bmatrix} L' \\ M' \\ S' \end{bmatrix}$$

For normalized XYZ input, the range of values attainable in this space is given by $I \in [0, 1]$ and $P, T \in [-1, 1]$. It would be possible to scale the axes by (100, 150, 150), making the color space compatible with, and roughly equivalent to, CIELAB (albeit with better hue uniformity).

The inverse conversion, between IPT and XYZ, is then given by the following set of equations:

$$\begin{bmatrix} L' \\ M' \\ S' \end{bmatrix} = \begin{bmatrix} 1.8502 & -1.1383 & 0.2384 \\ 0.3668 & 0.6439 & -0.0107 \\ 0.0000 & 0.0000 & 1.0889 \end{bmatrix} \begin{bmatrix} I \\ P \\ T \end{bmatrix}$$

$$L = \begin{cases} L'^{1/0.43} & \text{if } L' \geq 0, \\ -\left(-L'\right)^{1/0.43} & \text{if } L' < 0; \end{cases}$$

$$M = \begin{cases} M'^{1/0.43} & \text{if } M' \geq 0, \\ -\left(-M'\right)^{1/0.43} & \text{if } M' < 0; \end{cases}$$

$$S = \begin{cases} S'^{1/0.43} & \text{if } S' \geq 0, \\ -\left(-S'\right)^{1/0.43} & \text{if } S' < 0, \end{cases}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 1.0000 & 0.0976 & 0.2052 \\ 1.0000 & -1.1139 & 0.1332 \\ 1.0000 & 0.0326 & -0.6769 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix}$$

2.8 COLOR CORRECTION

Without the camera response function (Section 5.7), one cannot linearize the input as needed for color correction. Thus, a color correction value will not apply equally to all parts of the tone scale. For instance, darker colors may end up too blue compared with lighter colors. Furthermore, colors with primary values clamped to the upper limit (255 in an 8-bit image) have effectively been desaturated by the camera. While users are accustomed to this effect in highlights, after color correction such desaturated colors may end up somewhere in the midtones, where desaturation is unexpected. In a naïve method, whites may even be moved to some nonneutral value, which can be very disturbing.

Figure 2.24 shows the problem of color correction from a low dynamic range (LDR) original. If the user chooses one of the lighter patches for color balancing, the result may be incorrect due to clamping in its value. (The captured RGB values for the gray patches are shown in red.) Choosing a gray patch without clamping avoids this problem, but it is impossible to recover colors for the clamped patches. In particular, the lighter neutral patches end up turning pink in this example. The final image shows how these problems are avoided when an HDR original is available. Since the camera response curve has been eliminated along with clamping, the simple approach of balancing colors by choosing a neutral patch and multiplying the image by its inverse works quite well.

2.9 COLOR APPEARANCE

The human visual system adapts to the environment it is viewing (see Chapter 7 for more information). Observing a scene directly therefore generally creates a different visual sensation than observing an image of that scene on a (LDR) display. In the case of viewing a scene directly, the observer will be adapted to the scene. When looking at an image of a display, the observer will be adapted to the light emitted from the display, as well as the environment in which the observer is located.

There may, therefore, be a significant mismatch between the state of adaptation of the observer in these two cases. This mismatch may cause the displayed image to be perceived differently from the actual scene. The higher the dynamic range of the

(a) White balanced capture
(b) Off-balance capture
(c) LDR white balancing using second patch
(d) LDR white balancing using fourth patch
(e) HDR white balancing

FIGURE 2.24 (a) shows a Macbeth ColorChecker^{tm} chart captured with the appropriate white balance setting under an overcast sky. (b) shows the same scene captured using the "incandescent" white balance setting, resulting in a bluish color cast. (Red dots mark patches that cannot be corrected because one or more primaries is clamped to 255.) (c) shows an attempt to balance white using the second gray patch, which was out of range in the original. (d) shows the best attempt at correction using the fourth gray patch, which was at least in range in the original. (e) shows how range issues disappear in an HDR original, allowing for proper postcorrection.

scene, the larger this difference may be. In HDR imaging, and in particular in tone reproduction, it is, therefore, important to understand how human vision adapts to various lighting conditions and to develop models that predict how colors will be perceived under such different lighting conditions. This is the domain of color appearance modeling [82].

An appearance of color is influenced by various aspects of the viewing environment, such as the illuminant under which the stimulus is viewed. The chromatic adaptation transforms discussed in Section 2.5 are an important component of most color appearance models.

The color of the area surrounding the stimulus also plays an important role, as shown in Figure 2.25, where the same gray patch is shown on different

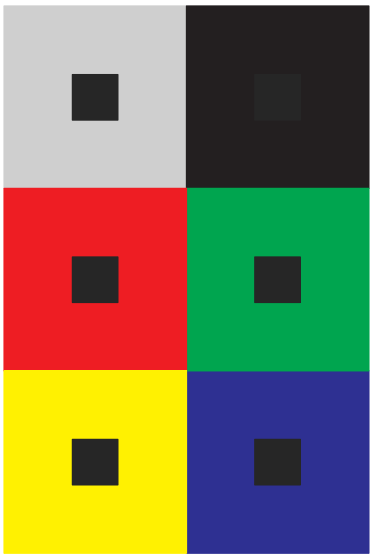


FIGURE 2.25 *Simultaneous color contrast shown for an identical gray patch displayed on differently colored backgrounds.*

backgrounds. The color of the patch appears different in each case — an effect known as "simultaneous color contrast."

To characterize a stimulus within a specific environment, first, its tristimulus value is specified in CIE XYZ color space. Second, attributes of the environment in which the stimulus is viewed need to be provided. If the stimulus is a homogeneous reflecting patch of color on a neutral (gray) background, this characterization of the environment may be as simple as the specification of an illuminant.

The appearance of a color is then described by "appearance correlates," which may be computed from the color's tristimulus values as well as the description of the environment [82]. Useful appearance correlates include lightness, chroma, hue, and saturation, which are defined later in this section. Our definitions are based on Wyszecki and Stiles's *Color Science* [372].

Brightness. The attribute of visual sensation according to which a visual stimulus appears to emit more or less light is called "brightness" and ranges from bright to dim.

Lightness. The area in which a visual stimulus is presented may appear to emit more or less light in proportion to a similarly illuminated area that is perceived as a white stimulus. Lightness is, therefore, a relative measure and may be seen as relative brightness. Lightness ranges from light to dark. In both CIELUV and CIELAB color spaces, L^* is the correlate for lightness. Note that if the luminance value of the stimulus is approximately 18% of Y_n, that is, $Y/Y_n = 0.18$, the correlate for lightness becomes approximately 50, which is halfway on the scale between light and dark. In other words, surfaces with 18% reflectance appear as middle gray. In photography, 18% gray cards are often used as calibration targets for this reason.[7]

Hue. The attribute of color perception denoted by red, green, blue, yellow, purple, and so on is called "hue." A chromatic color is perceived as possessing hue. An

7 Although tradition is maintained and 18% gray cards continue to be used, the average scene reflectance is often closer to 13%.

achromatic color is not perceived as possessing hue. Hue angles h_{uv} and h_{ab} may be computed as follows:

$$h_{uv} = \arctan \frac{v^*}{u^*}$$

$$h_{ab} = \arctan \frac{a^*}{b^*}$$

Chroma. A visual stimulus may be judged in terms of its difference from an achromatic stimulus with the same brightness. This attribute of visual sensation is called "chroma." Correlates of chroma may be computed in both CIELUV (C^*_{uv}) and CIELAB (C^*_{uv}):

$$C^*_{uv} = \left[(u^*)^2 + (v^*)^2 \right]^{1/2}$$

$$C^*_{ab} = \left[(a^*)^2 + (b^*)^2 \right]^{1/2}$$

Saturation. While chroma pertains to stimuli of equal brightness, saturation is an attribute of visual sensation, which allows the difference of a visual stimulus and an achromatic stimulus to be judged regardless of any differences in brightness. In CIELUV, a correlate for saturation s^*_{uv} may be computed as follows:

$$s^*_{uv} = \frac{C^*_{uv}}{L^*}$$

A similar correlate for saturation is not available in CIELAB.

Several more advanced color appearance models have recently appeared. The most notable among these are CIECAM97 [42,133,227,81], which exists in both full and simplified versions, and CIECAM02 [226,185]. As with the color spaces mentioned above, their use is in predicting the appearance of stimuli placed in a simplified environment. They also allow conversion of stimuli between different display media, such as different computer displays that may be located in different lighting environments. These recent color appearance models are generally more complicated than the procedures described in this section but are also deemed more accurate.

The CIECAM97 and CIECAM02 color appearance models, as well as several of their predecessors, follow a general structure but differ in their details. We outline this structure by example of the CIECAM02 model [226,185].

2.9.1 CIECAM02

The CIECAM02 color appearance model works under the assumption that a target patch with given relative tristimulus value XYZ is viewed on a neutral background and in the presence of a white reflective patch, which acts as the reference white (i.e., it is the brightest part of the environment under consideration). The background is again a field of limited size. The remainder of the visual field is taken up by the surround. This simple environment is lit by an illuminant with given relative tristimulus values $X_{\mathrm{W}}Y_{\mathrm{W}}Z_{\mathrm{W}}$. Both these relative tristimulus values are specified as input and are normalized between 0 and 100.

The luminance measured off the reference white patch is then assumed to be the adapting field luminance L_{a}—the only absolute input parameter, measured in cd/m^2.

The neutral gray background has a luminance less than or equal to the adapting field luminance. It is denoted Y_{b} and is specified as a fraction of L_{a}, also normalized between 0 and 100.

The final input to the CIECAM02 color appearance model is a classifier describing the surround as average, dim, or dark. This viewing condition parameter is used to select values for the intermediary parameters F, c, and N_c according to Table 2.5.

Further intermediary parameters n, N_{bb}, N_{cb}, and z are computed from the input as follows:

$$n = \frac{Y_{\mathrm{b}}}{Y_{\mathrm{W}}}$$

$$N_{\mathrm{cb}} = 0.725\left(\frac{1}{n}\right)^{0.2}$$

$$N_{\mathrm{bb}} = N_{\mathrm{cb}}$$

$$z = 1.48 + \sqrt{n}$$

Surround	F	c	N_c
Average	1.0	0.69	1.0
Dim	0.9	0.59	0.95
Dark	0.8	0.525	0.8

TABLE 2.5 *Values for intermediary parameters in the CIECAM02 model as a function of the surround description.*

Next, a factor F_{L} is computed from the adapting field luminance, which accounts for the partial adaptation to overall light levels:

$$k = \frac{1}{5\,L_{\mathrm{a}} + 1}$$

$$F_{\mathrm{L}} = 0.2\,k^4\,(5\,L_{\mathrm{a}}) + 0.1\left(1 - k^4\right)^2\,(5\,L_{\mathrm{a}})^{1/3} \tag{2.2}$$

The CIECAM02 color appearance model and related models proceed with three main steps, namely

- Chromatic adaptation
- Nonlinear response compression
- Computation of perceptual appearance correlates

The chromatic adaptation transform is performed in the CAT02 space, outlined in Section 2.5. The XYZ and $X_{\mathrm{W}}Y_{\mathrm{W}}Z_{\mathrm{W}}$ tristimulus values are first converted to this space:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = M_{\mathrm{CAT02}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Then, a degree of adaptation D is computed, which determines how complete the adaptation is. It is a function of the adapting field luminance, as well as the surround (through the parameters L_a and F):

$$D = F\left[1 - \frac{1}{3.6}\exp\left(\frac{-L_a - 42}{92}\right)\right]$$

The chromatically adapted signals are then computed:

$$R_c = R\left[\left(D\frac{Y_W}{R_W}\right) + (1 - D)\right]$$
$$G_c = G\left[\left(D\frac{Y_W}{G_W}\right) + (1 - D)\right]$$
$$B_c = B\left[\left(D\frac{Y_W}{B_W}\right) + (1 - D)\right]$$

After applying this chromatic adaptation transform, the result is converted back to XYZ space.

The second step of the CIECAM02 model is the nonlinear response compression, which is carried out in the Hunt–Pointer–Estevez color space, which is close to a cone fundamental space such as LMS (see Section 2.5). Conversion from XYZ to this color space is governed by the following matrix:

$$M_H = \begin{bmatrix} 0.3897 & 0.6890 & -0.0787 \\ -0.2298 & 1.1834 & 0.0464 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$$

The chromatically adapted signal after conversion to the Hunt–Pointer–Estevez color space is indicated with the $(R'G'B')$ triplet. The nonlinear response compression yields a compressed signal $(R'_aG'_aB'_a)$ as follows:

$$R'_a = \frac{400\ \left(F_L\ R'/100\right)^{0.42}}{27.13 + \left(F_L\ R'/100\right)^{0.42}} + 0.1 \tag{2.3}$$

$$G'_a = \frac{400\ \left(F_L\ G'/100\right)^{0.42}}{27.13 + \left(F_L\ B'/100\right)^{0.42}} + 0.1 \tag{2.4}$$

$$B'_a = \frac{400\ \left(F_L\ B'/100\right)^{0.42}}{27.13 + \left(F_L\ B'/100\right)^{0.42}} + 0.1 \tag{2.5}$$

This response compression function follows a "S"-shape on a log–log plot, as shown in Figure 2.26.

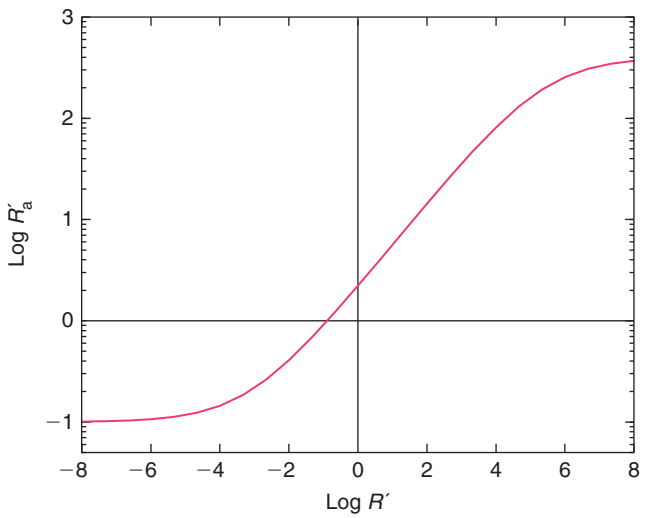


FIGURE 2.26 CIECAM02 nonlinear response compression on log–log axes.

The final step consists of computing perceptual appearance correlates. These describe the perception of the patch in its environment and include lightness, brightness, hue, chroma, colorfulness, and saturation. First, a set of intermediary parameters is computed, which includes a set of color opponent signals a and b, a magnitude parameter t, an achromatic response A, a hue angle h, and an eccentricity factor e:

$$a = R'_{\mathrm{a}} - 12\ G'_{\mathrm{a}}/11 + B'_{\mathrm{a}}/11$$

$$b = \left(R'_{\mathrm{a}} + G'_{\mathrm{a}} - 2\ B_{\mathrm{a}}\right)/9$$

$$h = \tan^{-}1(b/a)$$

$$e = \left(\frac{12500}{13}\ N_{\mathrm{c}}\ N_{\mathrm{cb}}\right)\left[\cos\left(\frac{h\pi}{180} + 2\right) + 3.8\right]$$

$$t = \frac{e\ \sqrt{a^2 + b^2}}{R'_{\mathrm{a}} + G'_{\mathrm{a}} + 21\ B'_{\mathrm{a}}/20}$$

$$A = \left[2\ R'_{\mathrm{a}} + G'_{\mathrm{a}} + B'_{\mathrm{a}}/20 - 0.305\right]\ N_{\mathrm{bb}}$$

The unique hues red, yellow, green, and blue have values for h as given in Table 2.6. The hue angles h_1 and h_2 for the two nearest unique hues are determined from the value of h and Table 2.6. Similarly, eccentricity factors e_1 and e_2 are derived from this table and the value of e. The hue composition term H_{i} of the next lower unique hue is also read off this table.

The appearance correlates may then be computed with the following equations, which are estimates for hue H, lightness J, brightness Q, chroma C, colorfulness M, and saturation s:

$$H = H_{\mathrm{i}} + \frac{100\ (h - h_1)/e_1}{(h - h_1)/e_1 + (h2 - h)/e_2}$$

$$J = 100\left(\frac{A}{A_{\mathrm{W}}}\right)^{cz}$$

Unique Hue	Hue Angle	Eccentricity Factor	Hue Composition
Red	20.14	0.8	0
Yellow	90.00	0.7	100
Green	164.25	1.0	200
Blue	237.53	1.2	300

TABLE 2.6 *Hue angles h, eccentricity factors e, and hue composition H_i for the unique hues red, yellow, green, and blue.*

$$Q = \left(\frac{4}{c}\right)\sqrt{\frac{J}{100}}\;(A_W + 4)\;F_L^{0.25}$$

$$C = t^{0.9}\sqrt{\frac{J}{100}}\;\left(1.64 - 0.29^n\right)^{0.73}$$

$$M = C\;F_L^{0.25}$$

$$s = 100\sqrt{\frac{M}{Q}}$$

These appearance correlates thus describe the tristimulus value XYZ in the context of its environment. Thus, by changing the environment only, the perception of this patch changes, and this will be reflected in the values found for these appearance correlates. In practice, this would occur for instance when an image displayed on a monitor and printed on a printer needs to appear the same. While colorimetry may account for the different primaries of the two devices, color appearance modeling additionally predicts differences in color perception due to the state of adaptation of the human observer in both viewing conditions.

If source and target viewing conditions are known, color appearance models may be used to convert a tristimulus value from one viewing condition to the other. The

first two steps of the model, chromatic adaptation and nonlinear response compression, would then be applied to, followed by the inverse of these two steps. During execution of the inverse model, the parameters describing the target environment (adapting field luminance, tristimulus value of the reference white, etc.) would be substituted into the model.

The field of color appearance modeling is currently dominated by two trends. The first is that there is a realization that the visual environment in which a stimulus is observed is in practice much more complicated than a uniform field with a given luminance. In particular, recent models are aimed at modeling the appearance of a pixel's tristimulus values in the presence of neighboring pixels in an image. Examples of models that begin to address these spatial configurations are the S-CIELAB and iCAM models [381,84,143,83,228]. A collection of color appearance models that can also be used for tone reproduction purposes is described in Section 8.2.

A second trend in color appearance modeling constitutes a novel interest in applying color appearance models to HDR data. In particular, there is a mismatch in adaptation of the human visual system in a typical scene where high-contrast ratios may be present and that of a human observer in front of a typical display device. Thus, if an accurate HDR capture of a scene is tone-mapped and displayed on a computer monitor, the state of adaptation of the human observer in the latter case may cause the scene to appear different from the original scene.

The iCAM "image appearance model," derived from CIECAM02, is specifically aimed at addressing these issues [143,83] and in fact may be seen as a tone-reproduction operator. This model is presented in detail in Section 8.2.2.

2.9.2 REFINEMENTS

It is reasonable to argue that the computational model used in CIECAM02 predominantly models photoreceptor behavior, although it includes notions of higher-level cognition as well. In particular, the degree of adaptation can be regarded as a higher-level concept. Nonetheless, chromatic adaptation and nonlinear compression can be thought of as mediated by photoreceptors. Clusters of photoreceptors can be modeled by the von Kries hypothesis, which states that they operate (to a large extent) independently from each other.

It is therefore unlikely that chromatic adaptation, a step which is carried out in CIECAM02 before the nonlinear response compression, should require a different color space, making the computations for each color channel dependent on the values recorded in the other two channels. In other words, CIECAM02 is not a strict von Kries model.

A recent refinement to CIECAM02 consists of merging the chromatic adaptation and nonlinear response compression steps into a single operation, carried out in LMS cone response space [169]. The key observation is that the semisaturation constant in the nonlinear response compression can be different for each of the three channels. In CIECAM02, the semisaturation constant is 27.13 for each channel (see Equation 2.3). This constant can be thought of as the level of adaptation of the observer. By choosing a triplet of constants, one for each channel, and basing these constants on the chosen white point, we are able to achieve chromatic adaptation and nonlinear response compression in a single operation. The semisaturation constants $(\sigma_R, \sigma_G, \sigma_B)^T$ are computed as follows:

$$\sigma_R = 27.13^{1/0.42}(D\,(R_W/100) + (1 - D))$$

$$\sigma_G = 27.13^{1/0.42}(D\,(G_W/100) + (1 - D))$$

$$\sigma_B = 27.13^{1/0.42}(D\,(B_W/100) + (1 - D)),$$

where $(R_W, G_W, B_W)^T$ is the white point and D is the degree of adaptation. The sigmoidal compression step is then given by

$$L' = 400\frac{(F_L L/100)^{0.42}}{(F_L L/100)^{0.42} + \sigma_L^{0.42}} + 0.1$$

$$M' = 400\frac{(F_L M/100)^{0.42}}{(F_L M/100)^{0.42} + \sigma_M^{0.42}} + 0.1$$

$$S' = 400\frac{(F_L S/100)^{0.42}}{(F_L S/100)^{0.42} + \sigma_S^{0.42}} + 0.1$$

A comparison can be made with corresponding color data sets, which have formed the basis of other chromatic adaptation transforms [184]. The results are shown in Figure 2.27 and indicate that the output of the combined chromatic

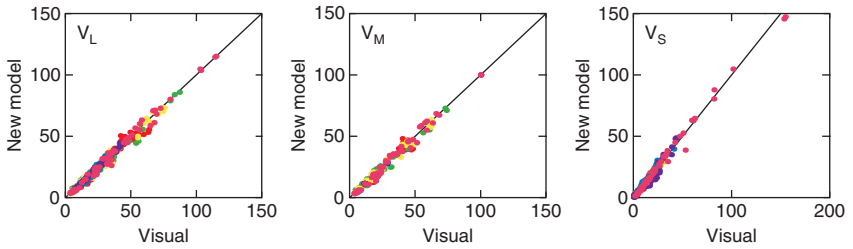


FIGURE 2.27 *The nonlinear response compression plotted against a collection of corresponding color data sets (indicated with "visual" in these plots).*

adaptation and nonlinear compression step is as accurate as the separate steps in CIECAM02.

A second refinement stems from the observation that color opponent spaces that decorrelate the data lead to red–green and yellow–blue channels, although these axes encode colors that are only approximately red, green, yellow, and blue. In fact, the opponent axes are closer to "pinkish red–cyan" and "greenish yellow–violet" [65]. The implication of this is that the color opponent encoding encountered in the ganglion cells in the retina does not directly correspond to the unique hues that are apparently computed in the visual cortex. This has led to a three-stage color model: LMS cone responses, ganglion color opponency, and a final recombination of the signal in terms of perceptually unique hues [55].

It has been shown that this three-stage color model is compatible with color appearance modeling, in that the model can be incorporated as the final step in a color appearance model, where it improves the prediction of appearance correlates [169].

In this model, the calculation of appearance correlates takes the aforementioned (L', M', S') triplet as input and derives the correlate of lightness J by means of the

achromatic signals SA and A_W as follows:

$$A = N_{\text{bb}}(4.19L' + M' + 1.17S')$$

$$A_{\text{W}} = N_{\text{bb}}(4.19L' + M' + 1.17S')$$

$$J = 106.5(A/A_{\text{W}})^{cz},$$

where the subscript W relates to the white point and the constants N_{bb}, c, and z are identical to those found in CIECAM02.

In this model, it is possible to compute chroma C independently from the hue angle using a color opponent pair $(a_{\text{c}}, b_{\text{c}})$, a normalization constant d, and the constants N_{c} and N_{cb} that are retained from CIECAM02. The computation requires an intermediary value t that is consequently simpler to compute than the CIECAM02 equivalent:

$$\begin{bmatrix} a_{\text{c}} \\ b_{\text{c}} \\ d \end{bmatrix} = \begin{bmatrix} -4.5132 & 3.9899 & 0.5233 \\ -4.1562 & 5.2238 & -1.0677 \\ 7.3984 & -2.3007 & -0.4156 \end{bmatrix} \begin{bmatrix} L' \\ M' \\ S' \end{bmatrix}$$

$$t = \frac{\left(N_{\text{c}} N_{\text{cb}} \sqrt{a_{\text{c}}^2 + b_{\text{c}}^2}\right)}{d}$$

$$C = (10^3 t)^{0.9} \sqrt{\frac{J}{100}} (1.64 - 0.29^n)^{0.73}$$

The computation of hue proceeds by computing a hue angle h by means of a second opponent space (a_h, b_h):

$$\begin{bmatrix} a_h \\ b_h \end{bmatrix} = \begin{bmatrix} -15.4141 & 17.1339 & -1.7198 \\ -1.6010 & -0.7467 & 2.3476 \end{bmatrix} \begin{bmatrix} L' \\ M' \\ S' \end{bmatrix} \quad h = \tan^{-1}\left(\frac{b_h}{a_h}\right)$$

To model the sharpening of the signal that occurs in the visual cortex [329], this intermediary hue angle is decomposed into perceptually unique hues for red, green, yellow, and blue. Note that the exponents on the cosines determine the amount of sharpening:

$$r_p = \max\left(0, 0.6581 \cos^{0.5390}(9.1 - h)\right)$$

$$g_p = \max\left(0, 0.9482 \cos^{2.9435}(167.0 - h)\right)$$

$$y_p = \max\left(0, 0.9041 \cos^{2.5251}(90.9 - h)\right)$$

$$b_p = \max\left(0, 0.7832 \cos^{0.2886}(268.4 - h)\right)$$

These values can then be encoded into an opponent space (a'', b'') by subtraction, and the final hue h' is then computed in the usual manner by taking the arctan of the ratio:

$$a'' = r_p - g_p$$

$$b'' = y_p - b_p$$

$$h' = \tan^{-1}\left(\frac{b''}{a''}\right)$$

The resulting appearance correlates can be validated against psychophysical data. The LUTCHI data set has been used to tune both CIECAM97 and CIECAM02 [198, 199]. For all relevant data in this test set, the root mean square (RMS) error for both CIECAM and the model presented in this section are given in Table 2.7. The result shows that the new model predicts all data better, but in particular, the computation of hue has improved significantly. The utility of this model therefore lies in its general ability to predict appearance correlates while solving the discrepancy between the orientation of conventional opponent axes and the perceptually unique hues.

Correlate	New Model	CIECAM02
Lightness	7.97	9.22
Chroma	8.68	9.04
Hue	12.40	21.70

TABLE 2.7 *The RMS error between the prediction of the color appearance models and the LUTCHI data for each of the lightness, chroma, and hue appearance correlates.*

2.10 DISPLAY GAMMA

Cathode ray tubes have a nonlinear relationship between input voltage V and light output L_{v}. This relationship is well-approximated with a power law function:

$$L_{\mathrm{v}} = k\ V^{\gamma}$$

The exponent γ models the nonlinearity introduced by the specific operation of the CRT and is different for different monitors. If V is normalized between 0 and 1, the constant k simply becomes the maximum output luminance L_{max}.

In practice, typical monitors have a gamma value between 2.4 and 2.8. However, further nonlinearities may be introduced by the lookup tables used to convert values into voltages. Thus, starting with a linear set of values that are sent to a CRT display, the result is a nonlinear set of luminance values. For the luminances produced by the monitor to be linear, the gamma of the display system needs to be taken into account. To undo the effect of gamma, the image data needs to be gamma-corrected before sending it to the display, as explained below.

Before the gamma value of the display can be measured, the black level needs to be set appropriately [256]. To set the black point on a monitor, first display a predominantly black image and adjust the brightness control on the monitor to its minimum. Then increase its value until the black image just starts to deviate from black. The contrast control may then be used to maximize the amount of contrast.

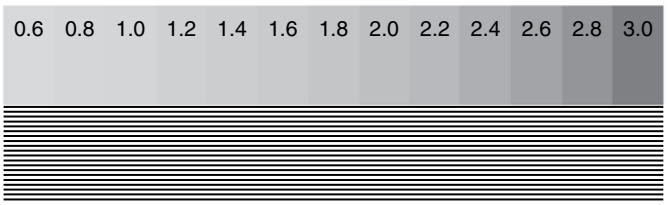


FIGURE 2.28 *Gamma estimation for CRT displays. The alternating black and white lines should be matched to the solid grays to determine the gamma of a display device.*

The gamma value of a display device may then be estimated by displaying the image shown in Figure 2.28. Based on an original idea by Paul Haeberli, this figure consists of alternative black and white lines on one side and solid gray patches on the other. By viewing this chart from a reasonable distance and matching the solid gray that comes closest to the gray formed by fusing the alternating black and white lines, the gamma value for the display device may be read off the chart. Note that this gamma-estimation chart should only be used for displays that follow a power-law transfer function, such as CRT monitors. This gamma-estimation technique may not work for LCD displays, which do not follow a simple power law.

Once the gamma value for the display is known, images may be precorrected before sending them to the display device. This is achieved by applying the following correction to the values in the image, which should contain normalized values between 0 and 1:

$$R' = R^{1/\gamma}$$

$$G' = G^{1/\gamma}$$

$$B' = B^{1/\gamma}$$

An image corrected with different gamma values is shown in Figure 2.29.

FIGURE 2.29 *An image corrected with different gamma values. In reading order:* $\gamma = 1.0$, 1.5, 2.0, *and* 2.5.

The technology used in LCD display devices is fundamentally different from CRT displays, and the transfer function for such devices is often very different. However, many LCD display devices incorporate circuitry to mimic the transfer function of a CRT display device. This provides some backward compatibility. Thus, while gamma encoding is specifically aimed at correcting for the nonlinear transfer function of CRT devices, gamma correction may often (but not always) be applied to images prior to display on LCD displays.

Many display programs perform incomplete gamma correction, that is, the image is corrected such that the displayed material is left intentionally nonlinear. Often, a gamma value of 2.2 is used. The effect of incomplete gamma correction is that contrast is boosted, which viewers tend to prefer [83]. In addition, display devices reflect some of their environment, which reduces contrast. Partial gamma correction may help regain some of this loss of contrast [350].

One of the main advantages of using a gamma encoding is that it reduces visible noise and quantization artifacts by mimicking the human contrast sensitivity curve.[8] However, gamma correction and gamma encoding are separate issues, as explained next.

2.11 BRIGHTNESS ENCODING

Digital color encoding requires quantization, and errors are inevitable during this process. In the case of a quantized color space, it is preferable for reasons of perceptual uniformity to establish a nonlinear relationship between color values and the intensity or luminance. The goal is to keep errors below the visible threshold as much as possible.

The eye has a nonlinear response to brightness — at most adaptation levels, brightness is perceived roughly as the cube root of intensity (see for instance the encoding of $L*$ of the CIELAB and CIELUV color spaces in Section 2.9). Applying a linear quantization of color values would yield more visible steps in darker regions than in the brighter regions, as shown in Figure 2.30.[9] A power law encoding with a γ value of 2.2 produces a much more even distribution of quantization steps, though the behavior near black is still not ideal. For this reason and others, some encodings such as sRGB add a short linear range of values near zero (see Section 2.12).

However, such encodings may not be efficient when luminance values range over several thousand or even a million to one. Simply adding bits to a gamma encoding does not result in a good distribution of steps, because it can no longer be assumed that the viewer is adapted to a particular luminance level, and the relative quantization error continues to increase as the luminance gets smaller. A gamma encoding

8 Rather than being constant, human contrast sensitivity depends on illumination [54].

9 We have chosen a quantization to 6 bits to emphasize the visible steps.

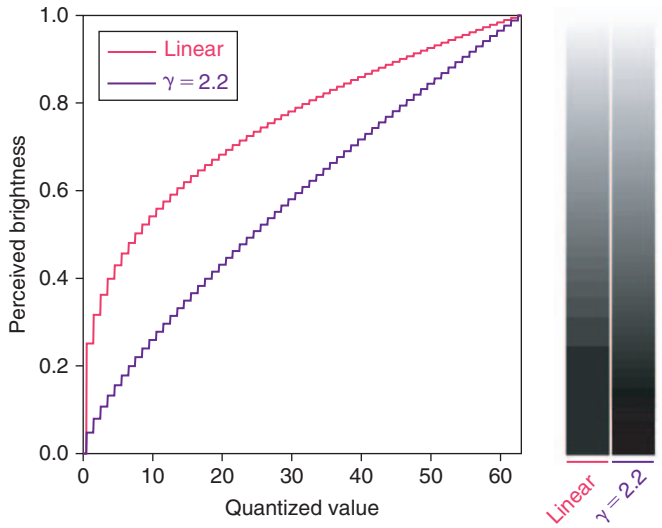


FIGURE 2.30 *Perception of quantization steps using a linear and a gamma encoding. Only 6 bits are used in this example encoding to make the banding more apparent, but the same effect takes place in smaller steps using 8 bits per primary.*

does not hold enough information at the low end to allow exposure readjustment without introducing visible quantization artifacts.

To encompass a large range of values when the adaptation luminance is unknown, an encoding with a constant or nearly constant relative error is required. A log encoding quantizes values using the following formula rather than the power law given earlier:

$$I_{\text{out}} = I_{\text{min}} \left[\frac{I_{\text{max}}}{I_{\text{min}}} \right]^{v}$$

This formula assumes that the encoded value v is normalized between 0 and 1 and is quantized in uniform steps over this range. Adjacent values in this encoding thus differ by a constant factor, equal to:

$$\left[\frac{I_{\max}}{I_{\min}}\right]^{1/N},$$

where N is the number of steps in the quantization. This is in contrast to a gamma encoding, whose relative step size varies over its range, tending toward infinity at zero. The advantage of constant steps is offset by a minimum representable value, $I_{\min}$, in addition to the maximum intensity we had before.

Another alternative closely related to the log encoding is a separate exponent and mantissa representation, better known as "floating point." Floating point representations do not have perfectly equal step sizes but follow a slight sawtooth pattern in their error envelope, as shown in Figure 2.31. To illustrate the quantization differences between gamma, log, and floating-point encodings, a bit size (12) and range (0.001–100) are chosen that can be reasonably covered by all three types. A floating-point representation with 4 bits in the exponent, 8 bits in the mantissa, and no sign bit is chosen, since only positive values are required to represent light.

By denormalizing the mantissa at the bottom end of the range, values between $I_{\min}$ and zero may also be represented in a linear fashion, as shown in this figure.[10] By comparison, the error envelope of the log encoding is constant over the full range, while the gamma-encoding error increases dramatically after just two orders of magnitude. Using a larger constant for γ helps this situation somewhat, but ultimately, gamma encodings are not well-suited to full HDR imagery where the input and/or output ranges are unknown.

2.12 STANDARD RGB COLOR SPACES

Most capture and display devices have their own native color space, generically referred to as "device-dependent RGB." While it is entirely possible to convert an

10 "Floating-point denormalization" refers to the linear representation of values whose exponent is at the minimum. The mantissa is allowed to have a zero leading bit, which is otherwise assumed to be 1 for normalized values, and this leads to a steady increase in relative error at the very bottom end, rather than an abrupt cutoff.

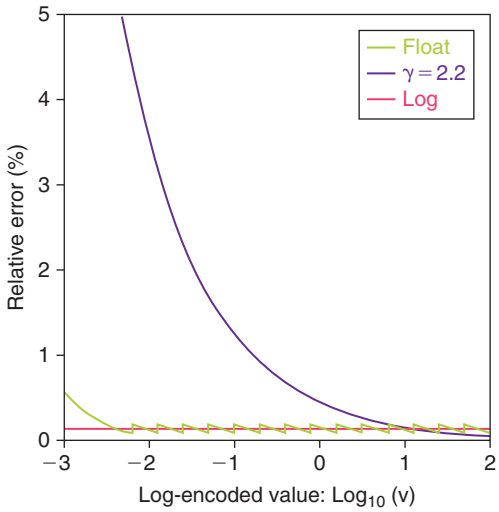


FIGURE 2.31 *Relative error percentage plotted against $\log_{10}$ of image value for three encoding methods.*

image between two device-dependent color spaces, it is more convenient to define a single standard color space that may serve as an intermediary between device-dependent color spaces.

On the positive side, such standards are now available. On the negative side, there is not one single standard but several competing standards. Most image encodings fall into a class called "output-referred standards," meaning they use a color space corresponding to a particular output device, rather than the original scene they are meant to represent. The advantage of such a standard is that it does not require any manipulation prior to display on a targeted device, and it does not "waste" resources on colors that are out of this device gamut. Conversely, the disadvantage of such a

standard is that it cannot represent colors that may be displayable on other output devices or may be useful in image-processing operations along the way.

A *scene-referred* standard follows a different philosophy, which is to represent the original, captured scene values as closely as possible. Display on a particular output device then requires some method for mapping the pixels to the device's gamut. This operation is referred to as "tone mapping" and may be as simple as clamping RGB values to a 0–1 range, or something more sophisticated, like compressing the dynamic range or simulating human visual abilities and disabilities (see Chapters 7 and 8). The chief advantage gained by moving tone mapping to the image decoding and display stage is that correct output can be produced for any display device, now and in the future. Also, there is the freedom to apply complex image operations without suffering losses due to a presumed range of values.

The challenge of encoding a scene-referred standard is finding an efficient representation that covers the full range of color values. This is precisely where HDR image encodings come into play, as discussed in Chapter 3.

For reference, we discuss several current output referenced standards. In Section 2.4, we already introduced the ITU-R RGB color space. In the remainder of this section, conversions to several other color spaces are introduced. Such conversions all follow a matrix multiplication followed by a nonlinear encoding. First, the sRGB color space is introduced as an example before generalizing the concept to other color spaces.

The nonlinear sRGB color space is based on a virtual display. It is a standard specified by the International Electrotechnical Commission (IEC 61966-2-1). The primaries and the white point are specified in terms of xy chromaticities according to Table 2.8 (this table also shows information for other color spaces that will be discussed momentarily). The maximum luminance for white is specified as 80 cd/m^2.

As the specification of sRGB is with respect to a virtual monitor, it includes a nonlinearity similar to gamma correction. This makes sRGB suitable for World Wide Web applications and "scanner-to-printer" applications. Many digital cameras now produce images in sRGB space. Because this color space already includes a nonlinear transfer function, images produced by such cameras may be displayed directly on typical monitors. There is generally no further need for gamma correction, except perhaps in critical viewing applications.

Color Space		Primaries			White Point (Illuminant)	
		R	G	B		
Adobe RGB (1998)	x	0.6400	0.2100	0.1500	D65	0.3127
	y	0.3300	0.7100	0.0600		0.3290
sRGB	x	0.6400	0.3000	0.1500	D65	0.3127
	y	0.3300	0.6000	0.0600		0.3290
HDTV (HD-CIF)	x	0.6400	0.3000	0.1500	D65	0.3127
	y	0.3300	0.6000	0.0600		0.3290
NTSC (1953)	x	0.6700	0.2100	0.1400	C	0.3101
	y	0.3300	0.7100	0.0800		0.3161
SMPTE-C	x	0.6300	0.3100	0.1550	D65	0.3127
	y	0.3400	0.5950	0.0700		0.3290
PAL/SECAM	x	0.6400	0.2900	0.1500	D65	0.3127
	y	0.3300	0.6000	0.0600		0.3290
Wide gamut	x	0.7347	0.1152	0.1566	D50	0.3457
	y	0.2653	0.8264	0.0177		0.3584

TABLE 2.8 *Chromaticity coordinates for primaries and white points defining several RGB color spaces.*

The conversion of CIE XYZ tristimulus values to sRGB consists of a 3×3 matrix multiplication followed by a nonlinear transfer function. The linear part of the transform is identical to the matrix specified in ITU-R Recommendation BT.709 introduced in Section 2.4. The resulting RGB values are converted into sRGB with the following transfer function (for R, G, and $B > 0.0031308$):

$$R_{\text{sRGB}} = 1.055\, R^{1/2.4} - 0.055$$
$$G_{\text{sRGB}} = 1.055\, G^{1/2.4} - 0.055$$
$$B_{\text{sRGB}} = 1.055\, B^{1/2.4} - 0.055$$

For values smaller than 0.0031308, a linear function is specified:

$$R_{\text{sRGB}} = 12.92\,R$$
$$G_{\text{sRGB}} = 12.92\,G$$
$$B_{\text{sRGB}} = 12.92\,B$$

This conversion follows a general pattern that is also found in other standards. First, a 3×3 matrix is defined, which transforms from XYZ to a color space with different primaries, and then a nonlinear transform is applied to the tristimulus values, which takes the following general form [245]:

$$R' = \begin{cases} (1+f)\,R^{\gamma} - f & \text{for } t \le R \le 1 \\ s\,R & \text{for } 0 < R < t \end{cases}$$

$$G' = \begin{cases} (1+f)\,G^{\gamma} - f & \text{for } t \le G \le 1 \\ s\,G & \text{for } 0 < G < t \end{cases}$$

$$B' = \begin{cases} (1+f)\,B^{\gamma} - f & \text{for } t \le B \le 1 \\ s\,B & \text{for } 0 < B < t \end{cases}$$

Note that the conversion is linear in a small dark region and follows a gamma curve for the remainder of the range. The value of s determines the slope of the linear segment, and f is a small offset. Table 2.9 lists several RGB standards, which are defined by their conversion matrices and their nonlinear transform specified by the γ, f, s, and t parameters [245]. The primaries and white points for each color space are given in Table 2.8. The gamuts spanned by each color space are shown in Figure 2.32. The gamut for the HDTV color space is identical to the sRGB standard and is therefore not shown again.

The Adobe RGB color space was formerly known as "SMPTE-240M" but was renamed after SMPTE's gamut was reduced. It has a larger gamut than sRGB, as shown in the chromaticity diagrams of Figure 2.32. This color space was developed with the printing industry in mind. Many digital cameras provide an option to output images in Adobe RGB color, as well as sRGB.

Color Space	XYZ to RGB Matrix	RGB to XYZ Matrix	Nonlinear Transform
Adobe RGB (1998)	$\begin{bmatrix} 2.0414 & -0.5649 & -0.3447 \\ -0.9693 & 1.8760 & 0.0416 \\ 0.0134 & -0.1184 & 1.0154 \end{bmatrix}$	$\begin{bmatrix} 0.5767 & 0.1856 & 0.1882 \\ 0.2974 & 0.6273 & 0.0753 \\ 0.0270 & 0.0707 & 0.9911 \end{bmatrix}$	$\gamma = \text{NA}$ $f = \text{NA}$ $s = \text{NA}$ $t = \text{NA}$
sRGB	$\begin{bmatrix} 3.2405 & -1.5371 & -0.4985 \\ -0.9693 & 1.8760 & 0.0416 \\ 0.0556 & -0.2040 & 1.0572 \end{bmatrix}$	$\begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix}$	$\gamma = 0.42$ $f = 0.055$ $s = 12.92$ $t = 0.003$
HDTV (HD-CIF)	$\begin{bmatrix} 3.2405 & -1.5371 & -0.4985 \\ -0.9693 & 1.8760 & 0.0416 \\ 0.0556 & -0.2040 & 1.0572 \end{bmatrix}$	$\begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix}$	$\gamma = 0.45$ $f = 0.099$ $s = 4.5$ $t = 0.018$
NTSC (1953)	$\begin{bmatrix} 1.9100 & -0.5325 & -0.2882 \\ -0.9847 & 1.9992 & -0.0283 \\ 0.0583 & -0.1184 & 0.8976 \end{bmatrix}$	$\begin{bmatrix} 0.6069 & 0.1735 & 0.2003 \\ 0.2989 & 0.5866 & 0.1145 \\ 0.0000 & 0.0661 & 1.1162 \end{bmatrix}$	$\gamma = 0.45$ $f = 0.099$ $s = 4.5$ $t = 0.018$

TABLE 2.9 *Transformations for standard RGB color spaces (after [245]).*

Color Space	XYZ to RGB Matrix	RGB to XYZ Matrix	Nonlinear Transform
SMPTE-C	$\begin{bmatrix} 3.5054 & -1.7395 & -0.5440 \\ -1.0691 & 1.9778 & 0.0352 \\ 0.0563 & -0.1970 & 1.0502 \end{bmatrix}$	$\begin{bmatrix} 0.3936 & 0.3652 & 0.1916 \\ 0.2124 & 0.7010 & 0.0865 \\ 0.0187 & 0.1119 & 0.9582 \end{bmatrix}$	$\gamma = 0.45$ $f = 0.099$ $s = 4.5$ $t = 0.018$
PAL/SECAM	$\begin{bmatrix} 3.0629 & -1.3932 & -0.4758 \\ -0.9693 & 1.8760 & 0.0416 \\ 0.0679 & -0.2289 & 1.0694 \end{bmatrix}$	$\begin{bmatrix} 0.4306 & 0.3415 & 0.1783 \\ 0.2220 & 0.7066 & 0.0713 \\ 0.0202 & 0.1296 & 0.9391 \end{bmatrix}$	$\gamma = 0.45$ $f = 0.099$ $s = 4.5$ $t = 0.018$
Wide gamut	$\begin{bmatrix} 1.4625 & -0.1845 & -0.2734 \\ -0.5228 & 1.4479 & 0.0681 \\ 0.0346 & -0.0958 & 1.2875 \end{bmatrix}$	$\begin{bmatrix} 0.7164 & 0.1010 & 0.1468 \\ 0.2587 & 0.7247 & 0.0166 \\ 0.0000 & 0.0512 & 0.7740 \end{bmatrix}$	$\gamma = \text{NA}$ $f = \text{NA}$ $s = \text{NA}$ $t = \text{NA}$

TABLE 2.9 (*continued*)

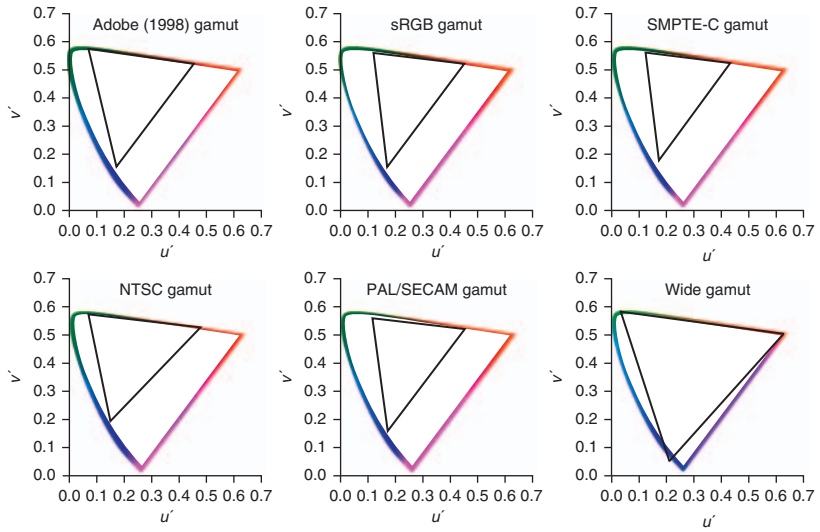


FIGURE 2.32 *CIE (u', v') chromaticity diagrams showing the color gamuts for various color spaces.*

The HDTV and sRGB standards specify identical primaries but differ in their definition of viewing conditions. As such, the difference lies in the nonlinear transform.

The NTSC standard was used as the color space for TV in North America. It has now been replaced with SMPTE-C to match phosphors in current display devices, which are more efficient and brighter. PAL and SECAM are the standards used for television in Europe.

Finally, the wide-gamut color space is shown for comparison [245]. Its primaries are monochromatic light sources with wavelengths of 450, 525, and 700 nm. This color space is much closer to the spectrally sharpened chromatic adaptation transforms discussed in Section 2.5.

This page intentionally left blank

High Dynamic Range Image Encodings

03

An important consideration for any digital image is how to store it. This is especially true for high dynamic range (HDR) images, which record a much wider gamut than standard 24-bit RGB, and therefore require an efficient encoding to avoid taking up an excess of disk space and network bandwidth. Fortunately, several HDR file encodings and formats have already been developed by the graphics community. A few of these formats have been in use for decades, whereas others have just recently been introduced to the public.

An *encoding* is defined as the raw bit representation of a pixel value, whereas a *format* includes whatever wrapper goes around these pixels to compose a complete image. The quality of the results is mostly determined by the encoding rather than the format, making encodings the focus of this chapter. File formats including some type of "lossy" compression are the exception to this rule because the image format must then be considered and evaluated as a whole. A few lossy HDR formats are now publicly available, and we offer a cursory comparison of these as well.

3.1 LOW- VERSUS HIGH DYNAMIC RANGE ENCODINGS

There is more than bit depth to defining the difference between HDR and low dynamic range (LDR) encodings. Specifically, a 24-bit RGB image is usually classified as an *output-referred* standard, because its colors are associated with some target output device. In contrast, most HDR images are *scene-referred* because their

© 2010, Elsevier Inc. All rights reserved.
DOI: 10.1016/B978-0-12-374914-7.00003-8

pixels have a direct relation to radiance in some scene, either real or virtual. This is logical because most output devices are low in dynamic range, whereas most scenes are high in dynamic range. One cannot refer a color encoding to scene values if those values are beyond what can be represented; thus, LDR images are inappropriate for scene-referred data. However, an HDR encoding *could* be used to hold output-referred data, but there would be little sense in it because scene-referred data can always be mapped to a particular output device, but not the reverse. A scene-referred-to-output-referred transformation is a one-way street for the simple reason that no output device can reproduce all that we see in the real world. This transformation is called "tone mapping," and is a topic we return to, frequently, in this book. (See Chapters 7 and 8.)

Having just introduced this standard term, it is important to realize that "scene-referred" is really a misnomer because no image format ever attempts to record *all* of the light projected from a scene. In most cases, there is little need for recording infrared and ultraviolet wavelengths, or completely sampling the visible spectrum, because the eye is *trichromatic*. As explained in Chapter 2, this means that it is sufficient to record three color channels to reproduce every color visible to a human observer. These may be defined by the CIE XYZ tristimulus space, or any equivalent three-primary space (e.g., RGB, YC_BC_R, CIELUV, etc.). As we are really interested in what people see, as opposed to what is available, it would be better to use the term "human-referred" or "perceptual" for HDR encodings.

Nonetheless, the term "scene-referred" is still preferred because sometimes we do wish to record more than the eye can see. Example applications for extrasensory data include the following:

- Satellite imagery, where the different wavelengths may be analyzed and visualized in false color.
- Physically based rendering, where lighting and texture maps interact to produce new colors in combination.
- Scientific and medical visualization, where (abstract) data is collected and visualized.

In such applications, we need to record more than we could see of a scene with the naked eye, and HDR formats are a necessary means to accomplish this. Further applications of HDR imagery are outlined in the following section.

3.2 APPLICATIONS OF HDR IMAGES

The demands placed on an HDR encoding vary substantially from one application to another. In an Internet application, file size might be the deciding factor. In an image database, it might be decoding efficiency. In an image-compositing system, accuracy might be most critical. The following are some of the many applications for HDR, along with a short list of their requirements.

Physically Based Rendering (Global Illumination): Perhaps the first application to use HDR images, physically based rendering and lighting simulation programs must store the absolute radiometric quantities for further analysis as well as perceptually based tone mapping [349,347]. In some cases, it is important to record more than what is visible to the human eye, as interactions between source and surface spectra multiply together. Additional accuracy may also be required of the encoding to avoid accumulated errors, and alpha and depth channels may also be desirable. A wide dynamic range is necessary for image-based lighting, especially in environments that include daylight. (See Chapter 11.)

Remote Sensing: As mentioned in the previous section, satellite imagery often contains much more than what is visible to the naked eye [187]. HDR is important for these images, but so is multispectral recording and the ability to annotate using image metadata. Accuracy requirements may vary with the type of data being recorded, and flexibility is the key.

Digital Photography: Camera makers are already heading in the direction of scene-referred data with their various RAW formats, but these are cumbersome and inconvenient compared with the standard encodings described in this chapter.[1] It is only a matter of time before cameras that write HDR images directly begin to appear in the market. File size is clearly critical to this application. Software compatibility is also important, although this aspect is largely neglected by camera RAW, although Adobe's Digital Negative (DNG) specification and software is an important step toward alleviating this problem [4].

1 Each camera manufacturer uses its own proprietary format, which is usually not compatible with other manufacturers' RAW formats or even with popular image-editing software. These formats are collectively called RAW because the camera's firmware applies only minimal processing to the data that is read off the sensor.

Image Editing: Image-editing applications with support for HDR image data are now available. Photoshop, since CS2, has included reading and writing of 32-bit pixel data, and supports more operations with each release, as does Photogenics (http://www.idruna.com), and the open-source application Cinepaint (http://www.cinepaint.org). A vast number of image-editing operations are possible on HDR data that are either difficult or impossible using standard output-referred data, such as adding and subtracting pixels without running under or over range, extreme color, contrast changes, and white balancing that works. Accuracy will be an important requirement here, again to avoid accumulating errors, but users will also expect support for all existing HDR image formats.

Digital Cinema (and Video): Digital cinema is an important and fast-moving application for HDR imagery. Currently, the trend is heading in the direction of a medium dynamic range, output-referred standard for digital film distribution. Film editing and production, however, will be done in some HDR format that is either scene-referred, or has some intermediate reference, such as movie film stock. For intermediate work, resolution and color accuracy are critical, but file size is also a consideration as there are more than 200 000 frames in a 2-h movie, and each of these may be composited from dozens of intermediate layers. Rendering a digital movie in HDR also permits HDR projection. (See Chapter 6 for details on display devices.) Looking further ahead, an exciting possibility is that HDR video may eventually reach the small screen. (See Chapter 4 for discussion on HDR video encodings.)

Virtual Reality: Many Web experiences require the efficient transmission of images, which are usually encoded as JPEG or some other lossy representation. In cases where a user is attempting to view or move around a virtual space, image exposure is often a problem. If there were a version of QuicktimeVR^{tm} that worked in HDR, these problems could be solved. Adopting standards for lossy HDR compression is, therefore, a high priority for VR on the Web.

Computer Games: Game engines are applying image-based rendering techniques (Chapter 11) and tone mapping for more lifelike experience, and standards are emerging for HDR texture compression as a critical element in this pipeline [230].

Each of these applications and HDR applications, not yet conceived, carry their own particular requirements for image storage. The following section lists and

compares the established HDR formats, and discusses upcoming formats toward the end.

3.3 HDR IMAGE FORMATS

Table 3.1 lists three existing HDR image formats and compares some of their key attributes. The encodings within these formats are broken out in Table 3.2, where the basic parameters are given. In some cases, one format may support multiple encodings (e.g., TIFF). The standard 24-bit RGB (sRGB) encoding is included as a point of comparison.

Formats based on logarithmic encodings, LogLuv24 and LogLuv32, maintain a constant relative error over their entire range.[2] For the most part, the floating-point encodings, RGBE, XYZE, IEEE RGB, and Half RGB, also maintain a constant relative error. The dynamic ranges quoted for sRGB is based on the point at which its relative

Format	Encoding(s)	Compression	Metadata	Support/Licensing
HDR	RGBE	run-length	calibration, color space,	open source software (Radiance)
	XYZE	run-length	+user-defined	easy implementation
TIFF	IEEE RGB	none	calibration, color space,	public domain library (libtiff)
	LogLuv24	none	+registered,	
	LogLuv32	run-length	+user-defined	
EXR	Half RGB	wavelet, ZIP	calibration, color space, +windowing, +user-defined	open source library (OpenEXR)

TABLE 3.1 *Established HDR image file formats.*

2 Relative step size is the difference between adjacent values divided by the value. The relative error is generally held to half the relative step size and is the difference between the correct value and the representation divided by the correct value.

Encoding	Color Space	Bits/pixel	Dynamic Range ($\log_{10}$)	Relative Error (%)
sRGB	RGB in [0,1] range	24	1.6 orders	variable
RGBE	positive RGB	32	76 orders	1.0
XYZE	(CIE) XYZ	32	76 orders	1.0
IEEE RGB	RGB	96	79 orders	0.000003
LogLuv24	$\log Y + (u', v')$	24	4.8 orders	1.1
LogLuv32	$\log Y + (u', v')$	32	38 orders	0.3
Half RGB	RGB	48	10.7 orders	0.1

TABLE 3.2 *HDR pixel encodings in the order of introduction.*

error passes 5%. Above 5%, adjacent steps in the encoding are easily distinguished. If one were to view an sRGB image on an HDR display, regions below 0.025 of the maximum would exhibit visible banding, similar to what is shown in Figure 3.1.[3] For luminance quantization to be completely invisible, the relative step size must be held less than 1% [372]. This is the goal of most HDR encodings, and some have relative errors considerably below this level. Pixel encodings with variable quantization steps are difficult to characterize in terms of their maximum dynamic range and are ill-suited for HDR applications in which the display brightness scaling is not predetermined.

3.3.1 THE HDR FORMAT

The "HDR" format, also known as the "Radiance" picture format (.hdr, .pic), was first introduced as part of the Radiance Lighting Simulation and Rendering System in 1989 [177] and has since found widespread use in the graphics community,

3 Thus, the dynamic range of sRGB is 0.025:1, which is the same ratio as $1{:}10^{1.6}$. In Table 3.2, we just report the number of orders (powers of 10).

FIGURE 3.1 *Banding as a result of quantization at the 5% level.*

FIGURE 3.2 *Bit breakdown for the 32-bit/pixel RGBE (and XYZE) encodings.*

particularly for HDR photography and image-based lighting [62,61]. (See Chapters 5 and 11.) The file wrapper consists of a short ASCII header, followed by a resolution string that defines the image size and orientation, followed by the run-length encoded pixel data. (Furthermore details and software are provided on the CD-ROM.) The pixel data comes in two flavors, a 4-byte RGBE encoding [342] and a CIE variant, XYZE. The bit breakdown is shown in Figure 3.2.

The RGBE components, R_M, G_M, and B_M, are converted from the scene-referred color (R_W, G_W, B_W) by the following formula:

$$E = \lceil \log_2 (\max (R_W, G_W, B_W)) + 128 \rceil$$

$$R_M = \left\lfloor \frac{256\, R_W}{2^{E-128}} \right\rfloor$$

$$G_M = \left\lfloor \frac{256\, G_W}{2^{E-128}} \right\rfloor$$

$$B_M = \left\lfloor \frac{256\, B_W}{2^{E-128}} \right\rfloor$$

There is also a special case for an input where $\max(R_W, G_W, B_W)$ is less than 10^{-38}, which is written out as (0,0,0,0). This gets translated to (0,0,0) on the reverse conversion. The reverse conversion for the normal case is:

$$R_W = \frac{R_M + 0.5}{256}\, 2^{E-128}$$

$$G_W = \frac{G_M + 0.5}{256}\, 2^{E-128}$$

$$B_W = \frac{B_M + 0.5}{256}\, 2^{E-128}$$

The conversions for XYZE are precisely the same, only with CIE X, Y, and Z substituted for R, G, and B. Because the encoding does not support negative values, using XYZE instead of RGBE extends the range to cover the entire visible gamut. (See Chapter 2 for details on the CIE XYZ space.) The dynamic range for these encodings is quite large, more than 76 orders of magnitude, and the accuracy is sufficient for most applications. Run-length encoding achieves an average of 25% compression (1:1.3), making the image files about as big as uncompressed 24-bit RGB.

3.3.2 THE TIFF FLOAT AND LOGLUV FORMATS

For over a decade, the Tagged Image File Format (.tif, .tiff) has included a 32-bit/component IEEE floating-point RGB encoding [3]. This standard encoding is in

some ways the ultimate in HDR image representations, covering nearly 79 orders of magnitude in miniscule steps. The flip side to this is that it takes up more space than any other HDR encoding—more than three times the space of the Radiance format described in the preceding section. Most TIFF libraries do not even attempt to compress this encoding, because floating-point data generally does not compress very well. Where one might get 30% compression from run-length encoding of RGBE data, 10% is what one can expect using standard entropy compression (e.g., ZIP) on the same data stored as IEEE floats. This is because the last 12 bits or more of each 24-bit mantissa will contain random noise from whatever camera or global illumination renderer generated them. There simply are no image sources with seven decimal digits of accuracy, unless they are completely synthetic.

Nevertheless, 96-bit/pixel RGB floats have a place, and that is as a lossless intermediate representation. TIFF float is the perfect encoding for quickly writing out the contents of a floating-point frame buffer, and reading it later without loss. Similarly, raw floats are a suitable means to send image data to a compositor over a high bandwidth local connection. It can also serve as a "gold standard" for evaluating different HDR representations, as shown in Section 3.4. However, most programmers and users are looking for a more compact representation, and within TIFF there are two: 24-bit and 32-bit LogLuv.

The LogLuv encoding was introduced as a perceptually based color encoding for scene-referred images [175]. Like the IEEE float encoding just described, LogLuv is implemented as part of the popular public-domain TIFF library. (The TIFF library and appropriate examples are included on the CD-ROM.) The concept is the same for the 24-bit and 32-bit/pixel variants, but they achieve a different range and accuracy. In both cases, the scene-referred data is converted to separate luminance (Y) and CIE (u, v) channels. (Review Chapter 2 for the conversions between CIE and RGB color spaces.) The logarithm of luminance is then taken, and the result is quantized into a specific range, which is different for the two encodings, although both reserve the 0 code for $\mathrm{Y} = 0$ (black). In the case of the 24-bit encoding, only 10 bits are available for the log-luminance value. Quantization and recovery are computed as follows:

$$L_{10} = \lfloor 64(\log_2 Y_{\mathrm{W}} + 12) \rfloor$$

$$Y_{\mathrm{W}} = 2^{\frac{L_{10}+0.5}{64} - 12}$$

This encoding covers a world luminance (Y_W) range of 0.00025:15.9, or 4.8 orders of magnitude in uniform (1.1%) steps. In cases where the world luminance is skewed above or below this range, we can divide the scene luminances by a constant and store this calibration factor in the TIFF "STONITS" tag.[4] When decoding the file, applications that care about absolute values consult this tag and multiply the extracted luminances accordingly.

The remaining 14 bits of the 24-bit LogLuv encoding are used to represent chromaticity, based on a look up of CIE (u, v) values as diagrammed in the lower portion of Figure 3.3. A zero lookup value corresponds to the smallest v in the visible gamut, and subsequent table entries are built up left to right, then bottom to top in the diagram. The uniform step size for u and v is 0.0035, which is just large enough to cover the entire visible gamut in 2^{14} codes. The idea is that employing a *perceptually uniform* color space, where equal steps correspond to equal differences in color,

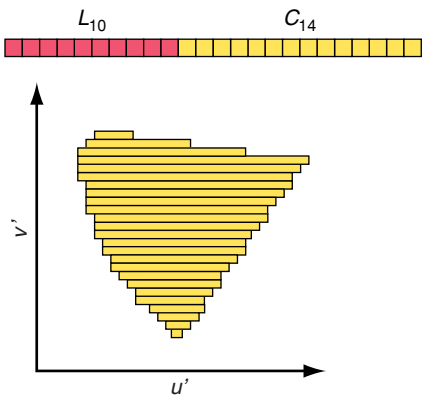


FIGURE 3.3 Bit breakdown for 24-bit LogLuv encoding and method used for CIE (u, v) lookup.

4 STONITS stands for "sample-to-nits." Recall from Chapter 2 that the term *nits* is shorthand for candelas per square meter.

keeps quantization errors below the visible threshold. Unfortunately, both the (u, v) step size and the luminance step size for the 24-bit encoding are slightly larger than the ideal. This quantization was chosen to cover the full gamut over a reasonable luminance range in a 24-bit file, and the TIFF library applies dithering to hide steps where they might otherwise be visible.[5] Because there is no compression for the 24-bit LogLuv encoding, there is no penalty in dithering.

The 32-bit LogLuv TIFF encoding is similar to the 24-bit LogLuv variant, but allows a greater range and precision. The conversion for luminance is as follows:

$$L_{15} = \lfloor 256\ (\log_2 Y_W + 64) \rfloor$$

$$Y_W = 2^{\frac{L_{15}+0.5}{256} - 64}$$

This 15-bit encoding of luminance covers a range of 5.5×10^{-20}:1.8×10^{19}, or 38 orders of magnitude in 0.3% steps. The bit breakdown for this encoding is shown in Figure 3.4. The leftmost bit indicates the sign of luminance, permitting negative values to be represented.[6] The CIE u and v coordinates are encoded in 8 bits each, which allows for sufficiently small step sizes without requiring a lookup.

FIGURE 3.4 *Bit breakdown for 32-bit LogLuv encoding. Upper and lower order bytes are separated per scan line during run-length compression to reduce the file size.*

5 Dithering is accomplished during encoding by adding a random variable in the (−0.5,0.5) range immediately before integer truncation.

6 This is useful for certain image-processing operations, such as compositing and error visualizations.

The conversion for chromaticity is simply as follows:

$$u_8 = \lfloor 410\, u' \rfloor$$

$$v_8 = \lfloor 410\, v' \rfloor$$

$$u' = \frac{u_8 + 0.5}{410}$$

$$v' = \frac{v_8 + 0.5}{410}$$

Again, dithering may be applied by the TIFF library to avoid any evidence of quantization, but it is not used for 32-bit LogLuv by default because the step sizes are below the visible threshold and run-length compression would be adversely affected. The compression achieved by the library for undithered output is 10–70%. Average compression is 40% (1:1.7).

Most applications will never see the actual encoded LogLuv pixel values, since the TIFF library provides conversion to and from floating-point XYZ scan lines. However, it is possible through the use of lookup on the raw encoding to combine the reading of a LogLuv file with a global tone-mapping operator (TM), thus avoiding floating-point calculations and providing for rapid display [174]. The TIFF library provides raw data access for this purpose.

The TIFF 16-bit/channel Format The TIFF 16-bit/channel (48 bits/pixel) encoding has been in use for quite some time, although it is not generally thought of as an HDR format. When pixels are encoded linearly, this may be true. However, using a gamma encoding of 2.2, 16 bits cover a perfectly ample range, with the 5% error level occurring seven orders of magnitude below the maximum representable value. We cannot represent negative primary values, so we are restricted to the gamut of whatever color space we are in, but this is true also of the Radiance RGBE format. The real problem is how this TIFF encoding is regarded by image-editing software such as Photoshop, where the maximum 16-bit value is equated to "white." This may be overcome by converting from 16-bit to 32-bit mode after reading the TIFF, then adjusting the exposure to suit. Just be careful when converting back to 16 bits not to clamp the highlights in the process. Proper exposure

usually requires the maximum value appear as a white spot in the image surrounded by what appear to be very dark pixels.

3.3.3 THE OPENEXR FORMAT

The EXtended Range format (.exr) was made available as an open source C++ library in 2002 by Industrial Light and Magic (see http://www.openexr.org) [144]. It is based on a 16-bit Half floating-point type, similar to IEEE float with fewer bits. Each RGB pixel occupies a total of 48 bits, broken into 16-bit words as shown in Figure 3.5. The Half data type is also referred to as "S5E10" for "sign, five exponent, ten mantissa." The OpenEXR library also supports full 32-bit/channel (96-bit/pixel) floats and a 24-bit/channel (72-bit/pixel) float type introduced by Pixar. Because it is the most widely used and supported, we restrict our discussion to the 16-bit/channel Half encoding.

The formula for converting from an encoded Half value is given below, where S is the sign bit, E the exponent (0–31) and M the mantissa (0–1023):

$$h = \begin{cases} (-1)^S \, 2^{E-15} \left(1 + \frac{M}{1024}\right) & 1 \leq E \leq 30 \\ (-1)^S \, 2^{-14} \, \frac{M}{1024} & E = 30 \end{cases}$$

If the exponent E is 31, then the value is either infinity if $M = 0$, or not a number (NaN). Zero is represented by all zero bits. The largest representable value in this encoding is 65504, and the smallest normalized (i.e., full-accuracy) value is

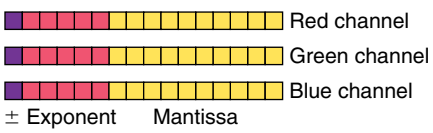


FIGURE 3.5 Bit breakdown for the OpenEXR Half pixel encoding.

0.000061. This basic dynamic range of nine orders is enhanced by the "denormalized" values less than 0.000061, which have a relative error less than 5% down to 0.000001, for a total dynamic range of 10.7 orders of magnitude. Over most of this range, the quantization step size is less than 0.1%, which is far below the visible threshold. This permits extensive image manipulations before artifacts become evident, which is one of the principal strengths of this encoding. Another advantage of the Half encoding is that the selfsame 16-bit representation is specified in NVidia's Cg language [211], as well as their newer CUDA [119].

The OpenEXR library contains C++ classes for reading and writing EXR image files, with support for lossless compression, tiling, and mip-mapping. Compression is accomplished using the ZIP deflate library as one alternative, or Industrial Light and Magic's (ILM) more efficient "PIZ" lossless wavelet compression. From our experiments, PIZ achieves a 60% reduction on average compared to uncompressed 48-bit/pixel RGB. OpenEXR also supports arbitrary data channels, including alpha, depth, and user-defined image data. Similar to the TIFF format, standard attributes are provided for color space, luminance calibration, pixel density, capture date, camera settings, and so forth. User-defined attributes are also supported, and unique to OpenEXR is the notion of a "display window" to indicate the active region of an image. This is particularly useful for special effects compositing, where the notion of what is on- and what is off-screen may evolve over the course of a project.

3.3.4 OTHER ENCODINGS

There are a few other encodings that have been used or are being used to represent "medium dynamic range" image data, that is, between two and four orders of magnitude. The first is the Pixar log encoding, which is available in the standard TIFF library along with LogLuv and IEEE floating point. This 33-bit/pixel encoding assigns each of 11 bits to red, green, and blue, using a logarithmic mapping designed to fit the range of movie film. The implementation covers about 3.8 orders of magnitude in 0.4% steps, making it ideal for film work, but marginal for HDR work. Few people have used this encoding outside of Pixar, and they have themselves moved to a higher-precision format. Another image standard that is even more specific to film is the Cineon format, which usually records logarithmic density in 10 bits/channel

over a 2.0 range (http://www.cineon.com). Although these 2.0 orders of magnitude may correspond to slightly more range once the film response curve has been applied, it does not qualify as an HDR encoding, and it is not scene-referred. Mechanically, the Cineon format will handle greater bit depths, but the meaning of such an extension has never been defined.

3.3.5 "LOSSY" HDR FORMATS

All of the HDR image formats we have discussed so far are *lossless* insofar as once the original scene values have been converted into the encoding, no *further* loss takes place during storage or subsequent retrieval. This is a desirable quality in many contexts, especially when an image is expected to go through multiple storage and retrieval steps (with possible manipulations) before reaching its final state. However, there are some applications where a *lossy* format is preferred, particularly when the storage costs are onerous or the final "predisplay" version of the scene-referred data is ready.

A particularly desirable quality in any new image format is some level of backwards-compatibility with existing software, and the introduction of HDR presents a unique opportunity in this regard. As we are attempting to store a superset of what most image viewers understand, we only need to record the extra data needed to remap the LDR image back to its HDR original. A general approach to this problem is described by Ward [348]. The earliest such method was published by Spaulding et al. at Kodak [302], who used a residual image to capture into a wider color space, storing the arithmetic difference between the JPEG sRGB image and their larger ROMM RGB gamut.

Dolby's JPEG–HDR Format Ward and Simmons developed a still image format that is backwards-compatible with the 8-bit JPEG standard [340,348]. This encoding method stores a tone-mapped image as a JPEG/JFIF file, smuggling restorative information in a metadata marker. Naïve applications ignore this marker as extraneous, but newer software can recover the full HDR data by recombining the encoded information with the tone-mapped image. In most cases, the additional metadata adds about 20% to the original JPEG size. By comparison, most lossless HDR formats require 16 times as much storage space as JPEG.

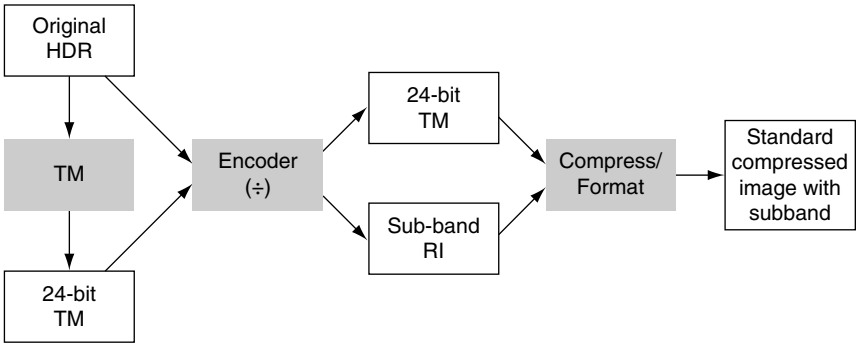


FIGURE 3.6 *The subband encoding pipeline. (Reprinted from [340])*

Figure 3.6 shows the encoding pipeline, including the to-be-specified TM. In principle, any TM can work, but the authors found that the photographic operator [274] (see Section 8.1.2) and the bilateral filter operator [74] (see Section 8.5.2) worked best with this method. Once the tone-mapped image is derived, its pixels are divided into the original to obtain a grayscale "ratio image," which is then compressed and incorporated in the JPEG file as a subband marker.

Figure 3.7 shows an HDR image of a church that has been decomposed into a tone-mapped image and the corresponding ratio image. The ratio image is downsampled and log encoded before being passed to the JPEG compressor to squeeze it into the metadata marker. Loss of detail is prevented either by enhancing edges in the tone-mapped image to compensate for the downsampled ratio image, or by synthesizing high frequencies in the ratio image during upsampling, depending on the application and user preference. These methods are called "precorrection" and "postcorrection," respectively. At the highest quality settings, the ratio image is recorded at full resolution, avoiding the need for these compensating measures.

The dynamic range of JPEG-HDR is unrestricted in the sense that the log encoding for the ratio image is optimized to cover the input range with the smallest step

FIGURE 3.7 *An HDR image of a church divided into a tone-mapped version and the downsampled ratio image that is stored as a subband.*

size possible. Furthermore, the color gamut is much wider than standard sRGB by virtue of an extended encoding of YCC values and the ability to *compand* saturation to maintain a natural appearance while preserving saturated colors [348].

Figure 3.8 illustrates the decoding process. A naïve application extracts the tone-mapped pixels and treats them as a standard output-referred image. An HDR application, however, recognizes the subband and decompresses both this ratio image and the tone-mapped version, multiplying them together to recover the original scene-referred data.

Two clear benefits arise from this strategy. First, a tone-mapped version of the HDR image is immediately available, for naïve applications that cannot handle anything more, and for HDR applications that may be able to perform their own

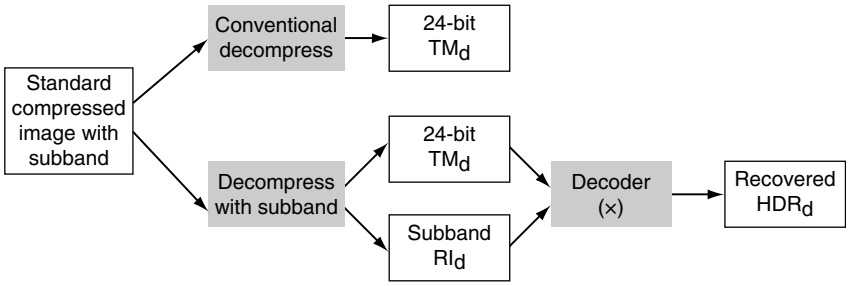


FIGURE 3.8 *The JPEG-HDR decoding pipeline. The lower left shows the backwards-compatible path for naïve applications, and the upper right shows the path for an HDR decoder.*

tone-mapping given time, but wish to provide the user with immediate feedback. Second, by making the format *backwards-compatible* with the most commonly used image type for digital cameras, an important barrier to adoption has been removed for the consumer, and hence for the camera manufacturers as well.

Ward and Simmons tested their encoding method on 15 different HDR images, passing them through a single encoding–decoding cycle and compared with the original using Daly's Visible Differences Predictor (VDP) [49]. With the exception of a single problematic image, VDP showed fewer than 0.2% of the pixels had visible differences at the maximum quality setting in any one image, and this went up to an average of 0.6% at a 90% quality setting.

The JPEG-HDR format is currently supported by Dolby as an adjunct to Tom Lane's original JPEG library. Tools for converting to and from JPEG-HDR are freely available, and the format is fully supported by the author's Photosphere browser and HDR image generator. A plug-in for Photoshop should be available at the time of this book's printing.

The XDepth Format Another HDR encoding that claims backwards-compatibility with JPEG is called XDepth (http://www.XDepth.com). Although details of the format are sketchy, a similar logic to JPEG-HDR is used, where a tone-mapped image

is stored in the standard location and metadata is included to reconstruct the HDR original. The tone mapping appears to be a simple gamma encoding, where optimal results are obtained by increasing the gamma to the maximum plug-in value of 4.0. This results in a somewhat faded appearance in the backwards-compatible image, but yields the most accurate HDR reconstruction. Compression performance with this setting is on par with JPEG-HDR, as can be seen in Figure 3.9. Both 12-Mpixel images were compressed to just 1.5 Mbytes, which is less than 5% of the original HDR file size. At this level of compression, artifacts are visible in darker regions of the recovered original, which we reexpose after reconstruction in the figure insets.

FIGURE 3.9 *The backwards-compatible formats JPEG-HDR and XDepth shown side by side. The outer images are what a naïve viewer would see for the two formats, respectively. The insets in the middle compare the reconstructed image for the problematic darker regions outlined in white. The center inset is the original HDR for comparison.*

3.3.6 MICROSOFT'S HD PHOTO FORMAT

With Windows Vista, Microsoft introduced the HD Photo image format (also known as Windows Media Photo). This is essentially a container format, similar in spirit to TIFF but with additional lossless and lossy compression methods competitive with JPEG 2000. Of particular interest here is the inclusion of Half, 32-bit/channel IEEE, and RGBE floating-point pixel types. Using HD Photo's lossless compression, it is possible to reduce files sizes by nearly a factor of two compared with Radiance's standard run-length encoding. Accepting some modest degradation, HD Photo's lossy compression can reduce this by a factor of four again, which brings it within striking range of JPEG-HDR and XDepth with comparable image quality. Of course, HD Photo lacks backwards-compatibility, but this is less important than optimizing for speed and application coverage in a new standard such as this.[7]

3.4 HDR ENCODING COMPARISON

To compare HDR image formats, we need a driving application. Without an application, there are no criteria, just speculation. The application determines the context and sets the requirements.

For this comparison, we have chosen a central application for HDR: scene-referred image archival. Specifically, we wish to store HDR images from any source to be displayed at some future date on an unknown device at the highest quality it supports. Assuming this display has not been invented, there is no basis for writing to an output-referred color space, hence a scene-referred encoding is the only logical representation. Thus, the need for HDR.

A reasonable assumption is that a full spectral representation is not necessary because humans perceive only three color dimensions. (Refer to Chapter 2.) Furthermore, we assume that it is not necessary to record more than the visible gamut, although it is not safe to assume that we can store less. Likewise, the quantization steps must be kept below the visible threshold, but since we plan

7 Due to some issues with the 1.0 release of the HD Photo plug-in for Photoshop CS3, we cannot offer a fair comparison to existing formats, although we expect these problems to be resolved in due course. What we find in our analysis of existing lossless formats below should largely apply to the same data types within HD Photo in its lossless mode.

no further manipulations prior to display, extra accuracy only means extra storage. The requirements for image archiving are therefore:

1. Cover the visible gamut with a tristimulus color space (XYZ, RGB, etc.)
2. Cover the full range of perceivable luminances
3. Have quantization steps below the visible threshold at all levels

Furthermore, it is desirable for the format to:

1. Minimize storage costs (megabytes/Mpixels)
2. Encode and decode quickly

Considering our requirements list (1–3 above), we can rule out the use of the RGBE encoding, which does not cover the visible gamut, and the 24-bit LogLuv encoding, which does not cover the full luminance range. This leaves us with the XYZE encoding (.hdr), the IEEE floating-point and 32-bit LogLuv encodings (.tif), and the Half encoding (.exr). Of these, the IEEE float representation will clearly lose in terms of storage costs, but the remaining choices merit serious consideration. These are as follows:

- The 32-bit Radiance XYZE encoding
- The 32-bit LogLuv encoding
- The 48-bit OpenEXR Half encoding

On the surface, it may appear that the XYZE and LogLuv encodings have a slight edge in terms of storage costs, but the OpenEXR format includes a superior compression engine. Also, the extra bits in the Half encoding may be worthwhile for some archiving applications that need or desire accuracy beyond normal human perception.

To evaluate the candidate formats, the following test was conducted on a series of IEEE floating-point images, some captured and some rendered:

1. The data is encoded into the test format, noting the central processing unit (CPU) time and disk space requirements.
2. The data is then decoded, noting the CPU time required.
3. The decoded data can then be compared with the original using CIE ΔE^* 1994 and gathered statistics.

CIE ΔE* 1994 is an updated version of the perceptual difference metric presented in Chapter 2 [216]. Using this metric, an encoded pixel color can be compared with the original source pixel by computing the visible difference. However, we must first modify the difference metric to consider local adaptation in the context of HDR imagery. To do this, the brightest Y value within a fixed region about the current pixel is found, and this value is used as the "reference white." This simulates the effect of a viewer adapting his or her vision (or display) locally, as we would expect him or her to do with an HDR image. The only question is how large a region to use, and for this a reasonable choice is to use a radius of 50 pixels, as this tends to be the size of interesting objects in our test image set.

Among the test images, we included a synthetic pattern that covered the full visible gamut and dynamic range with sufficient density to sample quantization errors at all levels. This pattern, a spiral slice through the visible gamut from 0.01 to 1 000 000 cd/m^2, is shown in Figure 3.10. (This image is included on the CD-ROM as an IEEE floating-point TIFF.) Each peak represents one revolution through the visible color gamut, and each revolution spans one decade (factor of 10) in luminance. The gray-looking regions above and below the slice actually contain random colors at each luminance level, which provide an even more thorough testing of the total space. Obviously, tone mapping has severely compressed the original dynamic range and colors to print this otherwise undisplayable image.

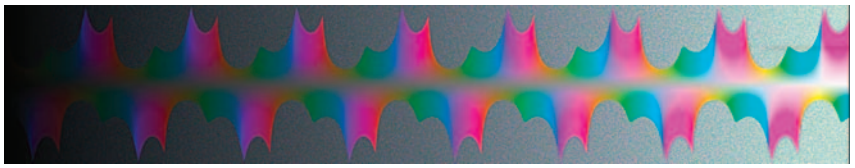


FIGURE 3.10 *The gamut test pattern, spanning eight orders of magnitude. This image was tone mapped with the histogram adjustment technique for the purpose of reproduction in this book (see Section 8.5.1).*

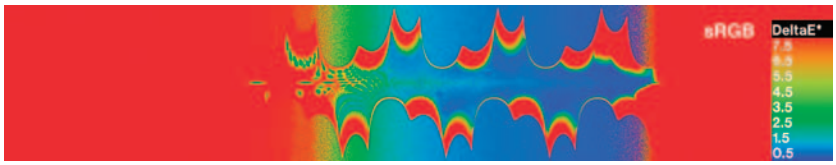


FIGURE 3.11 *This false color plot shows the visible error behavior of the 24-bit sRGB encoding on the test pattern shown in Figure 3.10. (CIE ΔE* values above 2 are potentially visible, and above 5 are evident.*

Figure 3.11 shows the CIE ΔE* encoding errors associated with a 24-bit sRGB file, demonstrating how ill-suited LDR image formats are for archiving real-world colors. Only a narrow region covering under two orders of magnitude with an incomplete gamut is below the visible difference threshold (2.0 in ΔE*). In contrast, the three HDR encodings we have chosen for this application do quite well on this test pattern, as shown in Figure 3.12. Errors are held below the visible threshold in each encoding over all eight orders, except for a few highly saturated colors near the top of the EXR Half range. The average ΔE* values for Radiance XYZE, 32-bit LogLuv, and EXR Half were 0.2, 0.3, and 0.06, respectively.

Figure 3.13 shows the two encodings we rejected on the basis that they did not cover the full dynamic range and gamut, and indeed we see they do not. As expected, the Radiance RGBE encoding is unable to represent highly saturated colors, although it easily spans the dynamic range. The 24-bit LogLuv encoding, however, covers the visible gamut, but only spans 4.8 orders of magnitude. Although they may not be well-suited to our proposed application, there are other applications to which these encodings are perfectly suited. In some applications, for example, there is no need to represent colors outside what can be displayed on an RGB monitor. Radiance RGBE has slightly better resolution than XYZE in the same number of bits and does not require color transformations. For other applications, 4.8 orders of magnitude may be sufficient because they only need to cover the human *simultaneous* range, that is, the range over which an observer can comfortably adapt without the use of blinders. Because 24-bit LogLuv covers the full gamut in this range as well, applications that

FIGURE 3.12 *Error levels for the chosen HDR encodings applied to the gamut test pattern from Figure 3.10, using the same scale as shown in Figure 3.11.*

need to fit the pixel data into a 24-bit buffer for historical reasons may prefer it to the 32-bit alternatives. For example, it was used in a proprietary hardware application where a prepared 24-bit lookup translated scene-referred colors to device space via a 16-million-entry table. Such a lookup would be wasteful with a 32-bit encoding, which would require four billion entries.

Besides color gamut and dynamic range, we are also interested in the statistical behavior of these formats on real images, especially with regard to file size and compression times. Figure 3.14 shows a test set of 34 images. Of these, 19 are HDR photographs of real scenes, and 15 are computer-generated, and sizes range from 0.2 to 36 Mpixels, with 2.4 Mpixels being average.

Figure 3.15 charts the read/write performance and file size efficiency for each of the three selected formats. This figure shows that the Radiance HDR format has the

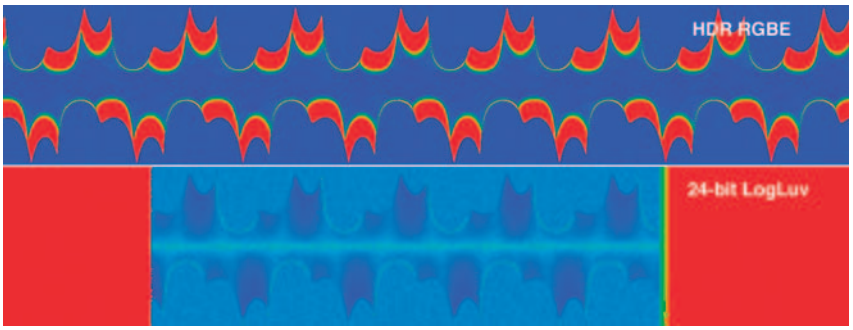


FIGURE 3.13 *The CIE ΔE* associated with the gamut test pattern in Figure 3.10 for the Radiance RGBE and 24-bit LogLuv encodings, using the same scale as Figure 3.11.*

fastest I/O performance, but creates larger files. The OpenEXR library is considerably slower with its I/O, but creates smaller files than Radiance, despite the 48 bits of the Half pixel encoding. The 32-bit LogLuv TIFF format has intermediate I/O performance, and produces the smallest files.

The average CIE ΔE* error performance of the three encodings is the same over the whole image set as we reported for the gamut test alone, with the following exceptions. One of the test images from ILM, "Desk," contained pixel values that are completely outside the visible gamut, and could not be reproduced with either Radiance XYZE or 32-bit LogLuv. As we do not expect a need for archiving colors that cannot be seen or reproduced, this should not count against these two encodings for this application. A few of the renderings had pixels outside the representable dynamic range of EXR's Half data type. In those cases, we did not resort to scaling the images to fit within the $10^{-6}:10^{4}$ range as we might have.

In summary, we found that the XYZE and LogLuv encodings are restricted to the visible gamut, and the Half encoding has a slightly smaller dynamic range. Neither of these considerations is particularly bothersome, so we conclude that all three encodings perform well for HDR image archiving.

FIGURE 3.14 *The test set of 34 HDR images.*

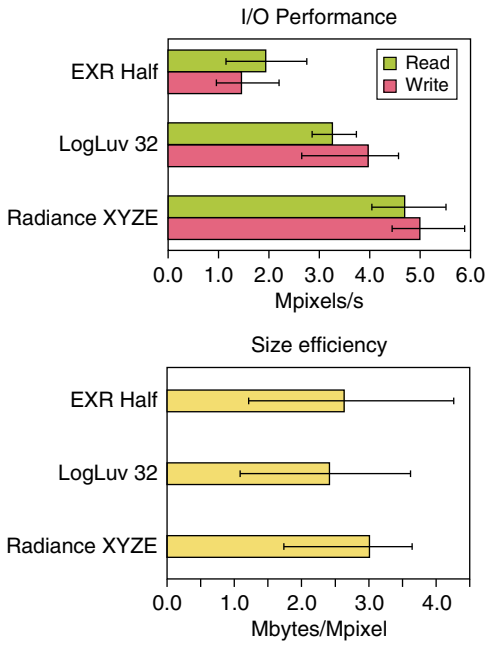


FIGURE 3.15 *I/O performance and file size for each of our three candidate formats. For I/O, longer bars mean faster reading and writing. Times were measured on a 1.8 GHz G5 processor. For size efficiency, shorter bars mean smaller files. Lines show the range of values (extrema) we saw in our tests.*

3.5 CONCLUSIONS

The principal benefit of using scene-referred, HDR images is their independence from the display process. A properly designed HDR format covers the full range

and sensitivity of human vision and is thus prepared for *any* future display technology intended for humans. Many HDR formats offer the further benefit, through additional range and accuracy, of permitting complex image operations without exposing quantization and range errors typical of more conventional LDR formats. The cost of this additional range and accuracy is modest—similar to including an extra alpha channel in an LDR format. This burden can be further reduced in cases where accuracy is less critical, that is, when multiple image read/edit/write cycles are not expected.

Most of the established HDR file formats are "lossless" in the sense that they do not lose information after the initial encoding, and repeated reading and writing of the files does not result in further degradation. However, "lossy" HDR formats are becoming more common, offering compression on par with existing JPEG images. This will remove an important barrier to HDR adoption in certain markets such as digital photography, video, and Web-based applications such as virtual reality tours.

High Dynamic Range Video Encodings

04

With increasing computing power, improving storage capacities, and the development of wireless and wired network systems, the role of video-based applications is expanding. Video has become common in multimedia messaging, video teleconferencing and telephony, video streaming, and TV signal broadcasting. Although in the near future, 8-bit-color-per-channel video-processing pipelines will continue to dominate in the vast majority of these applications, the demand for higher bit-depth capabilities is growing in video applications that are quality-demanding and precision-sensitive.

For example, imaging sensors in medical applications routinely produce image streams of 10-bit and higher precision, which is critical to maintain because of the demands of the application. In digital cinema workflows high dynamic range (HDR) frames are commonly used because of the lack of desirable contrast and luminance ranges as well as quantization errors, which are inherent in 8-bit signals. Furthermore, special effects in movies, especially those in which real and synthetic footage are seamlessly mixed, require both HDR shooting and rendering. HDR video is also required in all applications in which temporal aspects of changes in the scene must be captured with high accuracy. This is, in particular, important in monitoring of some industrial processes such as welding, predictive driver-assistance systems in the automotive industry, surveillance and remote sensing systems, to name just a few possible applications [127]. Because of HDR-enabled (floating-point) graphics pipelines, HDR image sequences can be readily generated as an output from modern Graphics Processing Unit (GPU) cards and potentially streamed over the network.

© 2010, Elsevier Inc. All rights reserved.
DOI: 10.1016/B978-0-12-374914-7.00004-X

In computer games and design applications, HDR video of real-world scenes can be used for virtual scene relighting [121] or as realistic video textures.

In all these applications, lossy video compression for high bit-depth signals is required to assure efficient video storage and transmission. That precludes the compression of each HDR frame independently (intraframe compression) as discussed in Section 3.3.5 without taking into account temporal coherence between frames (interframe compression). Fortunately, advances in video coding technology along with standardization efforts successfully address the demand for higher bit depth [371].

For example, the N-bit profile of MPEG-4 Part-2 Visual (often referred to as "MPEG-4 Visual") enables coding of luminance and chrominance signals up to 12 bits per color channel [136]. Also, in the most recent international video coding standard, H.264/AVC, its High 10 Profile supports 10-bit video, while the High 4:2:2 Profile can handle 12-bit coding [137]. All these formats allow frames to be encoded with higher precision, but they have not been specifically designed for HDR video.

What is needed for this purpose is the design of color spaces that can accommodate the full visible luminance range and color gamut, as discussed in Chapter 2. Then, after performing suitable quantization of HDR information encoded in such color spaces and mapping into integer values, one can use the existing standards for higher bit-depth encoding to actually store the HDR video. However, this creates another problem, as such custom encoding is not compliant with existing 8-bit decoders, precluding the reproduction of video on commonly used 8-bit display devices. In particular, backwards-compatibility should be maintained with encodings used by standard definition television (SDTV) or high-definition television (HDTV) such as H.264/AVS Main or High Profiles.

In recent years, we have witnessed a significant increase in the variation of display technology, ranging from sophisticated high-definition displays and digital cinema projectors to small displays on mobile devices. To accommodate video coding of the same content to all these devices, the scalable video coding (SVC) extension of H.264/AVS is proposed [283]. This extension enables the transmission and decoding of partial bit streams organized into layers, which are intended for each group of devices. At the decoding stage, the base layer common to all devices is combined with a number of enhancement layers to match the capabilities of a given display. Matching is achieved in terms of spatial resolution (number of pixels), temporal

resolution (desired frame rate), and image quality (referred also as "Signal-to-Noise Ratio (SNR) scalability").

In this context, bit-depth scalability is desirable as well, especially given that certain consumer displays already operate with 10-bit or higher signals. Furthermore, modern high-definition multimedia interfaces (HDMI v1.3) support 16-bit digital video as an optional encoding. In the current depth-scalable solutions [191], it is commonly assumed that the base coding layer is 8-bit (to capitalize on the best coding efficiency for 8-bit signals in existing codecs), while the additional enhancement layers (10-bits, 12-bits, or higher), when successively combined with the base layer, enable the reconstruction of the original HDR video. Note that by specifically selecting an 8-bit base layer encoding, which is compliant with H.264/AVS Main or High Profiles, backwards-compatibility can be achieved. However, HDR video can be decoded by any device aware of bit-depth enhancement layers, which can take advantage of such higher-precision video encoding.

At this point, an important distinction should be made between two possible approaches to pixel encoding in video frames. In HDR frames, pixel values are approximately linear to photometric luminance or radiometric radiance measures, whereas ready-to-display low dynamic range (LDR) frames undergo nonlinear gamma correction and tone mapping. Obviously, backwards-compatible frame encoding belongs to the second category. In our further discussion, we distinguish between two different types of solution. First, we deal with solutions in which HDR frames are transmitted to the display side where some custom processing is required. In this case, scene-referred encoding formats are considered, which enable additional rendering of a number of perceptual effects such as glare or motion blur (see Figure 4.6) that cannot be properly evoked in the human eye while observing LDR displays [164].

Second, it is possible to prebake ready-to-display material, possibly incorporating the transmission of higher-bit-depth frames. Here, display-referred encodings are performed, which may have clear advantages in terms of direct video reproduction on a given device. However, they may not be suitable for archival purposes. Mixed scenarios within the same video stream are clearly also possible, and thus the following three HDR video coding strategies can be distinguished:

Custom HDR Video Coding: A custom transform is used to convert floating-point HDR pixel color information into integers with a bit-depth per channel that is supported by the video coding. By applying the inverse transform, the original

floating-point values can be reconstructed (subject to quantization error) on the display side. Only a single-layer HDR video signal is encoded, and it is assumed that a custom tone mapping is performed by each display device. This way, an optimal LDR video for each display type can be obtained. In such conditions, backwards-compatibility with legacy devices is not assured. This strategy is the most efficient in terms of transmission bandwidth, but also requires encoding awareness and tone-mapping capabilities on the display side. Note that depending on the desired quality, tone reproduction on the display side can be an expensive operation.

Simulcast: Two or more nonscalable single-layer streams are transmitted (physically, those streams could be multiplexed into one stream), and each, possibly heterogeneous, client (e.g., display devices with different bit depths) selects the best-suited stream. Each stream is tuned (e.g., tone mapped) to its destination client and possibly does not require any further processing at the client side, except video decoding. This approach requires the largest transmission bandwidth, but usually leads to efficient backwards-compatible coding, and reduces the complexity of the decoder with respect to scalable approaches.

Scalable Bit-Depth Coding: Here, a single scalable multilayer stream is transmitted to heterogeneous clients, which require the same source content that is provided at different bit depths, spatial resolutions, frame rates, and bit rates (quality) [283]. The scalable stream is constructed to accommodate the highest possible requirements among clients, and the multilayer organization enables easy discarding of unwanted layers at the decoding stage. Such discarding could be also performed by Media-Aware Network Elements (MANEs) based on client characteristics and transmission channel capabilities before forwarding the adjusted video stream to its final destination. Ideally, scalable bit-depth coding can be performed in combination with other types of scalability: spatial, temporal, and quality [371]. At least the base layer is typically backwards-compatible and can handle ready-to-display tone-mapped video. Some extension layers could be designed to reconstruct the original HDR frames, subject to quantization error. Well-designed scalable approaches should have at most slightly worse encoding performance, and little increase in decoding complexity with respect to single-layer decoding, while at the same time possibly also adapting to the spatial resolution, and frame

and bit rates, of the display. The transmission bandwidth in scalable approaches should be significantly reduced with respect to simulcast, assuring comparable flexibility in terms of variety of video streams.

In the following sections, we provide examples of practical solutions for custom HDR video coding and backwards-compatible scalable bit-depth coding.

4.1 CUSTOM HDR VIDEO CODING

The MPEG-4 standard offers excellent performance of interframe coding, and it is natural that first attempts of lossy HDR video encoding aimed at capitalizing on this performance [205]. In particular, the Simple Profile ISO MPEG-4 (NOT_8_BIT) enables up to 12-bit-per-color-channel encoding. Thus, the main issue to address is how to encode 15 orders of magnitude of visible luminance range $[10^{-5}, 10^{10}]$ cd/m^2 into the available bit depth. As a linear encoding of such range would obviously lead to excessive quantization errors, an efficient color space is required that can accommodate all visible information in HDR. This issue is addressed further in Section 4.1.1. With the availability of such a color space, the remaining design decisions revolve around the choice of how this space can be used within the MPEG-4 framework. This is discussed in Section 4.1.2. Finally, in Section 4.1.3, we present an HDR video player that can take full advantage of HDR video in real time.

4.1.1 COLOR SPACE FOR LOSSY HDR PIXEL ENCODING

In designing a color space for the lossy encoding of HDR pixels [207], lessons from the construction of corresponding color spaces for LDR pixels should be taken. First of all, each color channel should be represented using positive integers, which leads to much better compression performance than floating-point representations. Also, redundancy of information between color channels should be minimized. For example, the color transform from RGB to the $\mathbf{YC}_B\mathbf{C}_R$ color space is performed at the initial stage of MPEG encoding to reduce strong correlations between channels inherent for RGB.

For HDR pixels, separation into one luminance and two chrominance channels leads to good results in nonlossy HDR image formats such as 32-bit LogLuv encoding

presented in Section 3.3.2. In HDR pixel encoding for video, a similar approach is taken. In particular, pixels are encoded into the CIE 1976 Uniform Chromaticity Scales u' and v' chromaticities, which are then multiplied by the factor 410. This rescales values to integer numbers that can be encoded into 8 bits. This approach of handling chromatic channels in HDR images was originally proposed by Ward [174], and it constitutes a good trade-off between compression efficiency and minimizing the visibility of possible color quantization artifacts.

The main issue that remains is the efficient encoding of luminance values into 12-bit integers that are supported by MPEG-4 and H.264/AVC. Here, the sensitivity of the human eye to contrast can be of great help in adjusting variable quantization steps that should always be below the visibility threshold for each luminance level. Some analogy can be found to the gamma nonlinearity in LDR image encoding (e.g., in the sRGB standard), but for HDR images, the full visible luminance range must be considered instead of the much narrower luminance ranges imposed by display technologies used in LDR imaging. Since the contrast sensitivity is significantly reduced for low luminance ranges up to roughly 10 cd/m^2, much higher quantization errors could be tolerated in dark image regions without introducing visible contouring artifacts. The dependency of threshold luminance on adaptation luminance is defined by the threshold-versus-intensity (TVI) function (see Figure 7.5), which plays a central role in efficient HDR luminance encoding.

The construction of a nonlinear transform from luminance L to integer luma $l_{\mathrm{HDR}}(L)$ is illustrated in Figure 4.1. Starting from the lowest considered luminance (in this case $L_{\min} = 10^{-5}\,\mathrm{cd/m^2}$) and assigning luma $l_{\mathrm{HDR}}(L_{\min}) = 0$, the origin of the transform curve is defined. Then, by finding the threshold luminance ΔL_1, which corresponds to adaptation luminance $L_{\min}$ using the TVI function (i.e., $\Delta L_1 = \mathrm{TVI}(L_{\min})$) the second point $l_{\mathrm{HDR}}(L_{\min} + \Delta L_1) = 1$ on the transform curve is created. In the same way, the threshold luminance ΔL_2 corresponding to $\mathrm{TVI}(L_{\min} + \Delta L_1)$ is found and the third point on the curve $l_{\mathrm{HDR}}(L_{\min} + \Delta L_1 + \Delta L_2) = 2$ is created. Any subsequent points can be found using $l_{\mathrm{HDR}}(L_{\min} + \Delta L_1 + \cdots + \Delta L_i) = i$ until $L_{\max} = 10^{10}\,\mathrm{cd/m^2}$ is reached.

Note that in this construction, we use luminance detection thresholds $\Delta L_1, \ldots, \Delta L_i$, so luma values are directly scaled in Just Noticeable Difference (JND) units. As $l_{\mathrm{HDR}}(L)$ is strictly monotonic, we can also find the inverse transform $L(l_{\mathrm{HDR}})$, which for each luma value l_{HDR} returns the corresponding luminance L. Then,

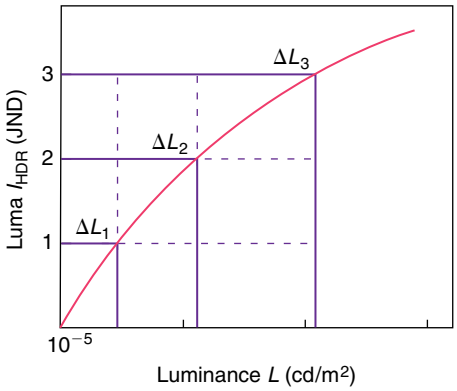


FIGURE 4.1 *Construction of a nonlinear luminance L to luma l_{HDR} mapping using luminance visibility thresholds $\Delta L_1, \ldots, \Delta L_i$, which are derived from the threshold-versus-intensity function $\Delta L_i = \mathrm{TVI}(L_i)$.*

the rounding error because of quantization to integer values should adhere to the following condition:

$$L(l_{\mathrm{HDR}} + 0.5) - L(l_{\mathrm{HDR}}) < \mathrm{TVI}(L_a), \tag{4.1}$$

where L_a is the adaptation luminance corresponding to the luma value l_{HDR}. Under this condition, the rounding error is below the visibility threshold, so that the resulting quantization artifacts cannot be seen. To have some safety margin, which reduces the likelihood of perceiving the quantization error, Mantiuk et al. [207] actually assume that:

$$L(l_{\mathrm{HDR}} + 0.5) - L(l_{\mathrm{HDR}}) < \frac{\mathrm{TVI}(L_a)}{1.4481} \tag{4.2}$$

Thus, for the unit difference in luma space l_{HDR}, we have:

$$L(l_{\mathrm{HDR}} + 1) - L(l_{\mathrm{HDR}}) < 2 \cdot \frac{1}{1.4481} \cdot \mathrm{TVI}(L_a) = 1.3811 \cdot \mathrm{TVI}(L_a) \tag{4.3}$$

This particular coefficient is chosen because then exactly 12 bits (i.e., 4096 luma values) are needed to encode the full visible luminance range $[10^{-5}, 10^{10}]$ cd/m^2) as originally assumed. This means that in the derivation of $l_{\mathrm{HDR}}(L)$ mapping, as illustrated in Figure 4.1, all luminance detection thresholds $\Delta L_1, \ldots, \Delta L_i$ should be rescaled by the coefficient 1.3812. Based on this assumption, Mantiuk et al. provide an analytic formula to encode HDR luminance L as 12-bit luma l_{HDR}:

$$l_{\mathrm{HDR}}(L) = \begin{cases} 769.18 \cdot L & \text{if } L < 0.061843 \text{ cd/m}^2 \\ 449.12 \cdot L^{0.16999} - 232.25 & \text{if } 0.061843 \leq L < 164.10 \text{ cd/m}^2 \\ 181.7 \cdot \ln(L) - 90.16 & \text{if } L \geq 164.10 \text{ cd/m}^2 \end{cases} \tag{4.4}$$

The inverse of this equation is given by:

$$L(l_{\mathrm{HDR}}) = \begin{cases} 0.0013001 \cdot l_{\mathrm{HDR}} & \text{if } l_{\mathrm{HDR}} < 47.568 \text{ cd/m}^2 \\ 2.4969 \cdot 10^{-16} \cdot (l_{\mathrm{HDR}} + 232.25)^{5.8825} & \text{if } 47.568 \leq l_{\mathrm{HDR}} < 836.59 \text{ cd/m}^2 \\ 1.6425 \cdot \exp(0.0055036 \cdot l_{\mathrm{HDR}}) & \text{if } l_{\mathrm{HDR}} \geq 836.59 \text{ cd/m}^2 \end{cases} \tag{4.5}$$

The actual shape of the mapping $l_{\mathrm{HDR}}(L)$ may depend slightly on the particular choice of the TVI function. The TVI function used in deriving Equations 4.4 and 4.5 is based on the contrast sensitivity function (CSF) as proposed by Daly [49] (see Section 10.7.3) for his visible difference predictor (VDP).[1] Figure 4.2 depicts the mapping $l_{\mathrm{HDR}}(L)$ based on Equation 4.4.

Another popular TVI function is rooted in experimental data measured by Blackwell [22], which is adopted as a CIE standard [41]. Figure 4.2 shows the corresponding $l_{\mathrm{HDR}}(L)$ mapping based on this TVI. For which analytic formula analogous to Equations 4.4 and 4.5 are provided in [207].

In Figure 4.2, we also show a logarithmic encoding that was fit to 12-bit luma. Note that with respect to two other encodings, many more bits have been allocated for dark image regions, which in fact is not needed due to lower contrast sensitivity

1 In Section 10.7.2, we discuss in more detail how the TVI function is actually derived from Daly's CSF.

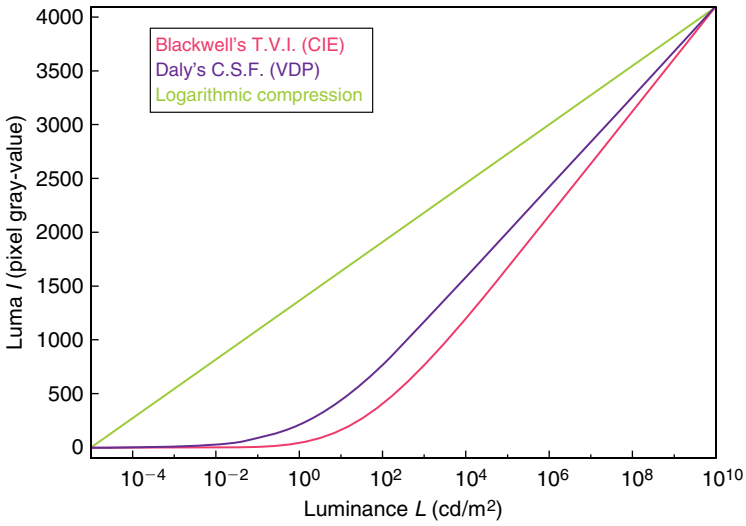


FIGURE 4.2 *Luminance-to-luma mappings for TVI functions proposed by Blackwell (pink) and Daly (purple). Also, a logarithmic function is included for comparison (green). (Image courtesy of Rafał Mantiuk)*

in such regions. However, for higher luminances the steepness of the logarithm function is lower, which means that the quantization errors are higher.

Figure 4.3 shows the comparison of quantization errors for the luma mapping presented in this section (in this case, for the CIE TVI function) with respect to other lossless HDR image formats (see Section 3.3). In all cases, the quantization error is below the visibility threshold, and therefore no artifacts because of luminance encoding should be visible. A major advantage of the $l_{\mathrm{HDR}}(L)$ mapping is that only 12 bits are used for encoding the full visible luminance range, whereas other formats usually require 16 bits for this purpose. This 4-bit reduction is possible mostly by taking into account the reduced contrast sensitivity for scotopic and mesopic

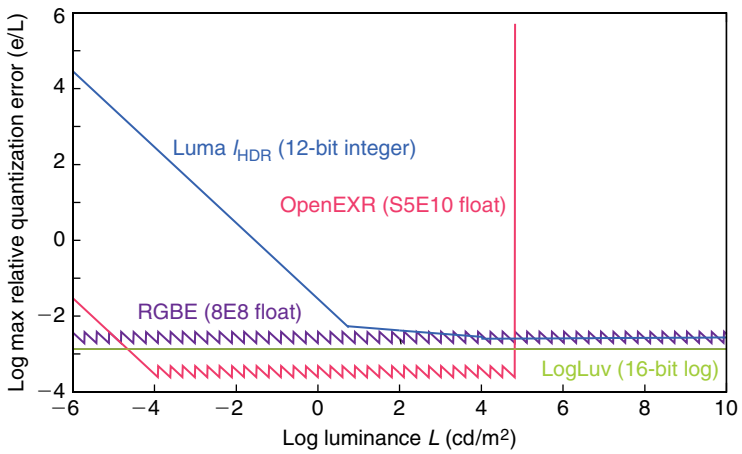


FIGURE 4.3 *Quantization error of popular luminance encoding formats: 12-bit integer* $l_{\mathrm{HDR}}(L)$ *encoding derived for the CIE TVI function, 16-bit floating-point OpenEXR format (see Section 3.3.3), 16-bit logarithmic encoding used in 32-bit version of LogLuv format (Section 3.3.2), 32-bit shared mantissa floating-point encoding used in Radiance's RGBE format (Section 3.3.1). The quantization error is computed as the maximum luminance rounding error* e *normalized by the corresponding luminance value* L*. (Image courtesy of Rafał Mantiuk)*

luminance levels. Note also that the RGBE and OpenEXR floating-point formats exhibit small, sawtooth-shaped fluctuations of the quantization error because of rounding of the mantissa. The OpenEXR format offers the highest precision in the range $[10^{-4}–10^{4.8}]$, and may have some problems reproducing luminance values for bright light sources.

Full advantage of the presented luma $l_{\mathrm{HDR}}(L)$ mapping can be taken provided that the encoded information is reasonably well-calibrated, and that the ultimate consumer is a human observer as opposed to a rendering algorithm in need of HDR input. These are defensible assumptions for compressed video. Even if the absolute calibration of the incoming luminances is unknown (for more details on

photometric camera calibration, refer to Section 7.6.1 or Chapter 3.2 of [232]), a suitable multiplier could be found that aligns luminance values in video with typical real-world luminance levels.

More details on this HDR pixel encoding is available elsewhere [205,203,207]. Note also that the DICOM (Digital Imaging and Communications in Medicine) standard [1], which is used in medical applications for grayscale image display, is based on principles similar to those used in the derivation of luma space $l_{\mathrm{HDR}}(L)$. However, the full visible luminance range as required in HDR applications cannot be accommodated by this DICOM standard. Similar limitations hold for other perceptually uniform quantization schemes [290,351,195], which are proposed mostly to reduce contouring artifacts. We refer the reader to these articles as they provide extensive perceptual background that is required for designing nonuniform quantization schemes with the smallest possible number of quantization levels.

This problem has been also studied in the context of digital cinema applications [314], in which direct encoding of pixels represented in CIE XYZ color space is considered. According to digital cinema standards, issued by the Digital Cinema Initiative (DCI), the following nonlinear transfer function is applied:

$$CX = \lfloor 4095 \times \left(\frac{X}{52.37}\right)^{\frac{1}{2.6}} \rfloor$$

$$CY = \lfloor 4095 \times \left(\frac{Y}{52.37}\right)^{\frac{1}{2.6}} \rfloor$$

$$CZ = \lfloor 4095 \times \left(\frac{Z}{52.37}\right)^{\frac{1}{2.6}} \rfloor, \tag{4.6}$$

which are encoded into 12-bit integer code values (CX, CY, CZ). Note that luminance Y, as well as the X and Z channels, are normalized by 52.37 cd/m^2, which produces a sufficiently large safety margin over the assumed reference white of 48 cd/m^2. For such relatively dark observation conditions, $\gamma = 1/2.6$ is assumed (refer to [256], Chapter 9, for an excellent discussion of gamma value selection as a function of ambient illumination and display luminance levels).

Psychophysical experiments confirm that such 12-bit encoding of luminance values falling into the [0.001, 50] cd/m^2 range is sufficient to avoid visible

quantization artifacts for a vast majority of participants [47]. Only 4–9% of all observers could properly distinguish the oriented square of waves superimposed on the uniform background. The amplitude of these waves is modulated in this case as a one-code value derived from Equation 4.6. Because the experiments were performed under extremely conservative test conditions (background uniformity, nonoverlapping stimuli, lack of noise, dithering, and grain), it was concluded that 12-bit encoding is sufficient for complex images, which usually exhibit substantial visual masking (see Section 10.7.5).

The choice of CIE XYZ color space offers many advantages in terms of explicit device independence (scene-referred encoding), backwards-compatibility, and extensibility to any other format. In addition, 12 bits are sufficient to encode the dynamic range up to 1:10 000 using Equation 4.6. However, such encoding is not optimal for real HDR content compression. In fact, as shown in this section, 12 bits are perfectly sufficient to cover the full $[10^{-5}, 10^{10}]$ cd/m^2 luminance range as perceived by a human observer, when contrast-sensitivity-aware encodings such as luma l_{HDR} are used.

4.1.2 MPEG-4 EXTENSION TO HDR VIDEO

The 12-bit luma $l_{\mathrm{HDR}}(L)$ encoding, as discussed in the previous section along with 8-bit u' and v' encodings, can be applied to the MPEG-4 framework [205]. In fact, in this work, an 11-bit encoding based on the same principles is shown to suffice. The scope of required changes with respect to standard LDR video encoding is surprisingly modest, as illustrated in Figure 4.4. As can be seen, all input frames are originally stored in the CIE XYZ color space, which enables the encoding of the full visible luminance range and color gamut. The luminance channel Y is directly converted into the 12-bit luma $l_{\mathrm{HDR}}(\mathrm{Y})$ integer encoding, while chromatic channels encoded using the CIE 1976 uniform chromaticity scales $u'v'$ have been adopted. The same $u'v'$ chromaticities are used in LogLuv encoding [174], as discussed in Section 3.3.2.

Note that the $u'v'$ chromaticities offer good perceptual uniformity, which in terms of encoding means that the potential quantization errors occur with similar magnitude for pixels with different colors. This simplifies the task of selecting the minimal number of bits required to avoid such quantization errors. It turns out

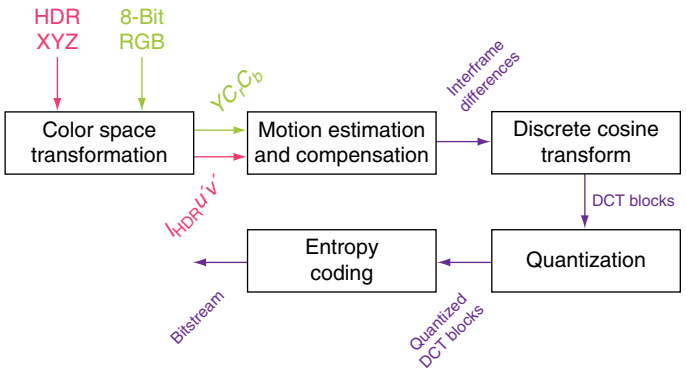


FIGURE 4.4 *Simplified pipeline for standard MPEG-4 video encoding (black) and its extension to handle HDR video (pink).*

that, in this case, 8 bits per channel is sufficient [174]. Actually, Mantiuk et al. [207] have observed barely visible contouring artifacts for blue and purple colors at the highest luminance levels on a good-quality calibrated liquid crystal display (LCD), but for complex images these artifacts are likely to be masked.

Another advantage of the $l_{\mathrm{HDR}}(\mathrm{Y})u'v'$ color space is the successful separation of chromatic and achromatic channels, which is known as the "isoluminance property of chroma channels." Such separation reduces redundancy of information among all channels and improves overall compression. This particular encoding replaces the standard color transform from RGB to the gamma-corrected $\mathrm{YC_BC_R}$ color space in MPEG-4, which does not exhibit such good isoluminance properties. This is discussed in more detail in Section 4.2.2, and in particular in the discussion accompanying Equations 4.7 and 4.8.

Because the N-bit profile of MPEG-4 Part-2 Visual enables the coding of pixel colors up to 12 bits per color channel [136], incorporating 12-bit luma $l_{\mathrm{HDR}}(\mathrm{Y})$ and 8-bit chroma $u'v'$ is relatively straightforward.

The remaining stages in the MPEG-4 compression pipeline (Section 6.5 in [25]), such as motion estimation and compensation, per-block discrete cosine transforms (DCT), quantization of the resulting DCT coefficients, and variable-length encoding of quantized coefficients, do not require any further customization to accommodate HDR video signals. This does not come as a surprise, as the vast majority of these operations are purely mathematical and do not show much awareness of the processed signal.

The only exception is the quantization step, which is responsible for information loss and, in fact, enables more aggressive compression by discarding all DCT coefficients that because of quantization are rounded to zero. For each DCT basis function that determines the spatial frequency of the encoded signal, a unique quantization step is specified in the so-called "quantization matrices," which differ between achromatic and chromatic color channels. In MPEG-4, the quantization matrices are tuned specifically for the YC_BC_R color space and luminance ranges typical for LDR display devices. See, for instance, Section 6.3.3 in [136], where the default quantization matrices used in MPEG-4 are specified.

In fact, the quantization matrices rely heavily on contrast perception characteristics of the human visual system, as discussed in Section 10.7.3. The rationale here is to use larger quantization steps for higher spatial frequencies, where the human visual system is less sensitive, as predicted by the CSF. Figure 10.9 shows the CSF shape as function of spatial frequency. Although the quantization matrices are designed for LDR display devices, the shape of the CSF changes only slightly for adaptation luminance levels more than 100 cd/m^2, as predicted by Weber's law.

This means that contrast sensitivity does not change for higher luminance levels, and the quantization steps specified in the standard quantization matrices can be safely applied to HDR material. As the contrast sensitivity is substantially lower for very low luminance levels, as can be seen in the plot for adaptation luminance $L_a = 0.1\,\text{cd/m}^2$ in Figure 10.9, we may expect better compression of HDR signals for custom-designed quantization matrices.

Although MPEG-4 supports the use of custom quantization matrices, they should be encoded into the stream. For each DCT block, the appropriate matrix should then be specified so that at the video decoding stage the correct inverse is selected. This creates additional storage overhead.

Another solution supported by the MPEG-4 standard uses per-block multipliers to rescale all quantization coefficients in the matrix. In this case, the per-block overhead is relatively modest, as usually 5 bits are used to identify the appropriate multiplier. These bits are stored only when there is a change of multipliers for neighboring blocks as signaled by an additional single bit code [279]. Clearly, further investigation is required to decide on the best strategy for encoding HDR frames, as they feature a large variation of luminance levels.

4.1.3 AN HDR VIDEO PLAYER

With increasing intelligence and computing power of many modern display devices, accommodating HDR video decoding and playing capabilities becomes more and more feasible. Providing display devices directly with HDR video signals opens many new possibilities with respect to currently used fully prebaked LDR frames. First, custom tone mapping can be performed by taking full advantage of the capabilities of the display in terms of brightness and contrast reproduction [164]. Second, ambient lighting conditions as well as other factors, such as the projected image size and observer distance, can be taken into account [202].

Note that for global tone-mapping operators aiming at a perceptual match between the appearance of real world scenes and displayed images, a small lookup table may be sufficient to drive the display device based on integer codes of the HDR video signal. Since the l_{HDR} encoding ensures that the quantization errors should not be visible under real-world observation conditions, they should not be visible in the tone-mapped images as well because of contrast compression [92,347]. However, local tone-mapping operators may amplify local image details beyond the perceptual visibility threshold in the real world, and quantization artifacts may therefore become apparent.

HDR video contains full luminance information about the scene even if it cannot be displayed using advanced tone mapping applied globally to the entire frame. Figure 4.5 shows a sequence of frames captured in the darkroom when at a certain moment a search light and then a strong halogen light are switched on. Although global tone mapping applied by the HDR video player can appropriately convey an overall impression about the scene, image details within the strong light sources cannot be seen. As such, information can be captured by HDR cameras and

encoded into HDR video; an interested user could explore such image details by selecting the luminance range in the scene that will be mapped to the full display luminance range. This is shown in the inset windows in Figure 4.5. Such local bright details would normally be clipped when LDR video is provided to the user.

Proper luminance calibration in the video stream enables us to simulate several perceptual effects adequate for specific luminance levels. Such effects are usually associated with tone mapping, and they must be rendered into each frame, as they

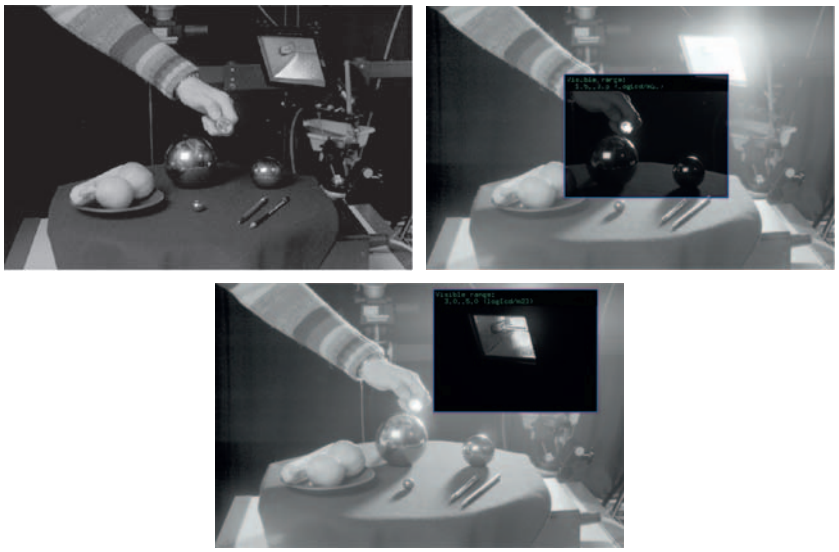


FIGURE 4.5 *Grayscale frames selected from a video captured using an HDR camera and encoded into an HDR MPEG-4 stream as described in Section 4.1.2. The blue frames not visible inset windows demonstrate that it is possible to explore the full visible luminance range by manipulating local exposure. (Images courtesy of Grzegorz Krawczyk)*

cannot be naturally triggered in the human eye while observing typical LDR displays. Some of these effects we discuss in the context of postprocessing for rendering with image-based lighting (Section 11.2.5), where the goal is to reproduce the optical imperfections of a real camera system, include motion blur (Figure 11.6), blooming (Figure 11.7), vignetting (Figure 11.8), and others. The vast majority of these effects do not require any knowledge about the three-dimensional scene representation and can, therefore, be readily reproduced through two-dimensional image-processing operations over rendered HDR frames. This means that they can be easily added atop HDR video as well.

Another luminance level effect that cannot be simulated on traditional LDR displays is night vision [92], which can be characterized by lower visual acuity, desaturated colors, and a bluish cast (Figure 4.6 left). In this case, simple Gaussian filtering and color manipulations can be used to render night vision effects [164]. Note that modern displays are getting much better in reproducing black levels with luminances below 0.01 cd/m^2 [287], in which case, scotopic vision effects such as the Purkinje shift can be naturally observed when such displays are placed in dark ambient light conditions. The Purkinje shift is the phenomenon that blue

FIGURE 4.6 *The appearance of night vision as an example of the perceptual effects that can be simulated in real time while playing an HDR video with scene-referred pixel encoding. Note the loss of visual acuity and color perception (left) with respect to the original daylight appearance (right). (Images courtesy of Grzegorz Krawczyk)*

patches appear brighter than red, although in daylight conditions the opposite effect is observed.

These examples show the potential of providing HDR video signals to the end user in terms of possible quality enhancement and appearance customization. Krawczyk et al. [164] present an integrated GPU framework, which reproduces all the effects discussed earlier, including local photographic tone mapping [274] in real-time while playing HDR video.

4.2 BACKWARDS-COMPATIBLE HDR VIDEO COMPRESSION

When observing the history of developments in broadcasting television signals, and more recently digital video, it is obvious that gradual evolution rather than revolution in the transition toward HDR video content distribution is to be expected. Furthermore, it is unclear to what extent content publishers will be willing to distribute fully HDR content. Another question relates to content owners' willingness to allow their material to be compromised on the display side. Content is normally color-graded, that is, adjustments have been applied in terms of tone and gamut mapping, allowing high-quality reproduction on typical CRT, LCD, and plasma display devices. This carefully tuned content could potentially be adjusted beyond the original intent by unskilled users who might decide to take full advantage of customization possibilities offered by HDR video representation (Section 4.1.3).

The current trend is that in response to the huge diversification of display devices, from tiny cell phone screens to digital cinema projectors, media publishers rely on color grading and video targeting for each platform, before distributing the resulting content by means of simulcast. In the following, we focus, therefore, on the technological aspects behind HDR content distribution.

Here, the most important issue is compliance with existing standards and backwards-compatibility with LDR video players and display devices. Such backwards-compatibility should be achieved at the expense of only moderate data overhead required to reconstruct HDR information. This is particularly important from the standpoint of LDR display users, who cannot directly benefit from HDR image quality, but still must bear additional storage or transmission costs.

To address these problems, it is possible to use tiny residual images that expand the dynamic range and color gamut inherent for 8-bit encoding [302,204]. Spaulding et al. use residual images to extend the dynamic range in sRGB encoding by an additional 2.2 orders of magnitude and demonstrate backwards-compatibility of their solution with the JPEG standard.

However, such relatively modest dynamic range extension may not be sufficient in HDR applications that often require at least six orders of magnitude. To this end, residual video streams may be embedded as a backwards-compatible MPEG4 extension to HDR video [204,206]. The international standardization efforts of the Joint Video Team (JVT) of ISO/IEC JTC1/SC29/WG11 (MPEG) and the Video Coding Expert Group (VCEG) of ITU-T SG16 Q.6 led to the development of scalable bit-depth coding solutions, which also rely on coding the residues between LDR and HDR layers [366,371,289,191].

Note that Section 3.3.5 details a backwards-compatible HDR JPEG solution [340, 348], which uses ratio images rather than residual images. These ratio images are then multiplied by the tone-mapped LDR image to restore the HDR original. Because the ratio image is usually downsampled and stores only luminance values, the reconstructed HDR image must rely on the LDR image with respect to spatial details and color information. This imposes some restrictions on tone-mapping operators and color spaces that are used in the construction of the LDR image. For example, precorrection compensating for losses of spatial information may be required, and clamping in bright image regions may remove color information. Such problems are mostly avoided with residual images, which store achromatic and chromatic channels at full resolution.

In the following section, we focus on video bit-depth expansion based on residual streams [204]. In Section 4.2.2, we then discuss how this concept can be extended for bit-depth scalable solutions embedded into the H.264/AVC framework.

4.2.1 HDR MPEG-4 VISUAL

Backwards-compatibility of MPEG-4 requires that the 8-bit LDR video stream is encoded without any changes with respect to its original color grading as performed by content providers. This is a strict requirements, and any departing from it cannot be tolerated, even if the LDR stream precorrection could potentially be beneficial for more efficient HDR video encoding [340,186].

As discussed in Chapter 9, inverse tone mapping could be used to reconstruct the original HDR content based on the tone-mapped LDR version. Such reconstruction is lossy because of quantization errors resulting from transforming floating-point or higher-bit-depth values into an 8-bit integer representation in the tone-mapping step. Such quantization errors may manifest themselves in reconstructed HDR video as visible banding artifacts.

It is even more difficult to reconstruct HDR image details in saturated (clipped) black-and-white image regions, where information is simply lost. Overall, however, inverse tone mapping leads to a reasonable prediction of HDR frames. This means that the difference between the predicted and original HDR frames contains mostly very small residuals, which can be efficiently compressed. The computation of such residual difference frames is trivial at the encoding stage when all required information is known. Embedding such a compressed residual stream into the MPEG stream enables almost perfect reconstruction of the original HDR frames at the decoding stage.

Figure 4.7 summarizes the encoding stage, which takes as input LDR and HDR frames. The decorrelation step corresponds to the computation of the residual video stream, which can also be interpreted as removing from the HDR frames all information that can be predicted from the LDR frames by performing inverse tone mapping. As mentioned earlier, residual frames tend to contain low-amplitude signals. This means that, when added back to the inverse tone-mapped LDR frames, they might not contribute to any perceivable changes in the appearance of the frame. Thus, filtering of such invisible details (essentially invisible noise) can be performed. Thus, the residual signal that is below the visibility threshold, as predicted by the achromatic and chromatic CSFs (see Section 10.7.3), is filtered out. This reduces the size of MPEG-encoded residual stream, which stores frames that are quantized to 8-bit residual.

The final MPEG-4 stream contains the LDR and residual streams as well as an auxiliary stream with the inverse tone-mapping function (encoded as a lookup table) and the quantization factors applied for the residual frame conversion into 8-bit representation. The auxiliary stream is encoded using lossless arithmetic.

Note that to derive the inverse tone-mapping transform, which is fully defined for the achromatic channel by a lookup table with 256 entries, the knowledge of the actual tone mapping used for LDR frame computation is not needed. By having access to both HDR and LDR frames at the encoding stage, the correspondence

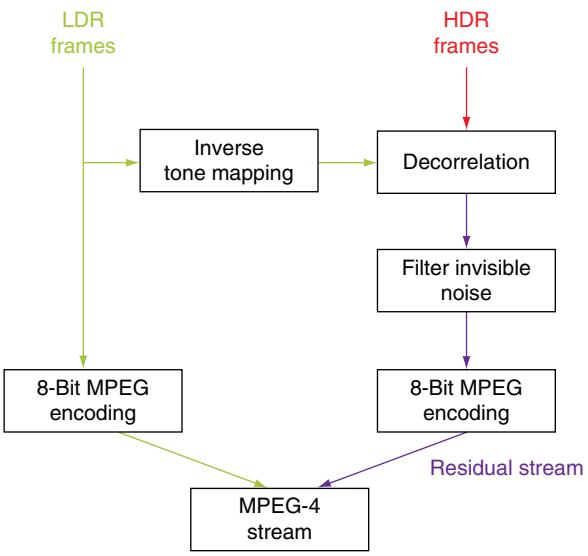


FIGURE 4.7 *Backwards-compatible extension of MPEG4 to HDR video using the stream of residual frames encoded along with traditional LDR video.*

between luma values for each pixel is known. In fact, it is a mapping of many HDR luma values that fall into one of 256 bins, and by computing their average value, a one-to-one lookup table between HDR and LDR luma values can be built. This lookup table can be used directly for inverse tone mapping, which allows HDR frames to be reconstructed based on its compressed LDR counterpart. The best prediction results are obtained when the reconstruction lookup table is built for each compressed frame independently.

The best prediction is possible for global tone-mapping operators, which are fully defined by a simple monotonic function, where the variance of HDR luma values falling into a single bin is merely because of the quantization error that is relatively small. This means that the error between the prediction and original HDR luma is

also small and can often be rounded down to zero as the result of the invisible noise-filtering step (Figure 4.7). Such filtering results in a much better compression of the residual stream.

For local tone mapping, which takes into account neighboring pixel intensities, similar HDR luma values may correspond to quite different bins, dependent on spatial location. This affects the prediction accuracy, and results in a substantially larger residual stream. This problem can be addressed successfully in scalable bit-depth coding solutions, which are discussed in Section 4.2.2. Figure 4.8 illustrates the relation of prediction lookup table to the actual HDR luma values distribution for a selected HDR frame. Wider distributions of HDR luma for each of 256 bins results in a worse prediction and poorer overall compression. For example, for the gradient domain operator proposed by Fattal et al. [89], which strongly enhances local contrast at the expense of possible distortions in global contrast reproduction, the prediction is particularly poor.

The overhead of the residual and auxiliary streams over the standard LDR video encoding is reported to be usually less than 30% [204].

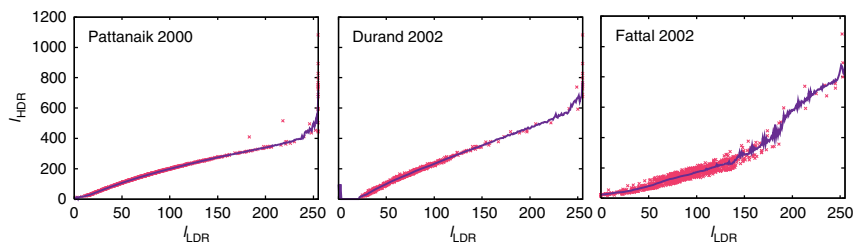


FIGURE 4.8 *The relation between LDR and HDR luma values for different tone-mapping operators (marked in pink) and the corresponding reconstruction lookup tables (marked in purple). From left to right: Pattanaik et al.'s global operator [248], Durand and Dorsey's local operator [74], and Fattal et al.'s local operator [89]. The data shown is for the Memorial Church image. (Plots courtesy of Rafał Mantiuk)*

4.2.2 SCALABLE BIT-DEPTH CODING IN THE H.264/AVC FRAMEWORK

SVC requires a bit-depth expansion that can be achieved by adding an enhancement layer that complements the LDR base layer, providing the missing HDR information. Figure 4.9 shows an example of a scalable encoder architecture, where the dashed line splits the encoder parts into an 8-bit base layer and a 10-bit enhancement layer (this specific bit depth is provided as an example). First, the 10-bit input signal is tone mapped to 8 bits, which is then encoded as a base layer using a single-layer encoder. The 8-bit base layer should be decodable by any nonscalable single-layer

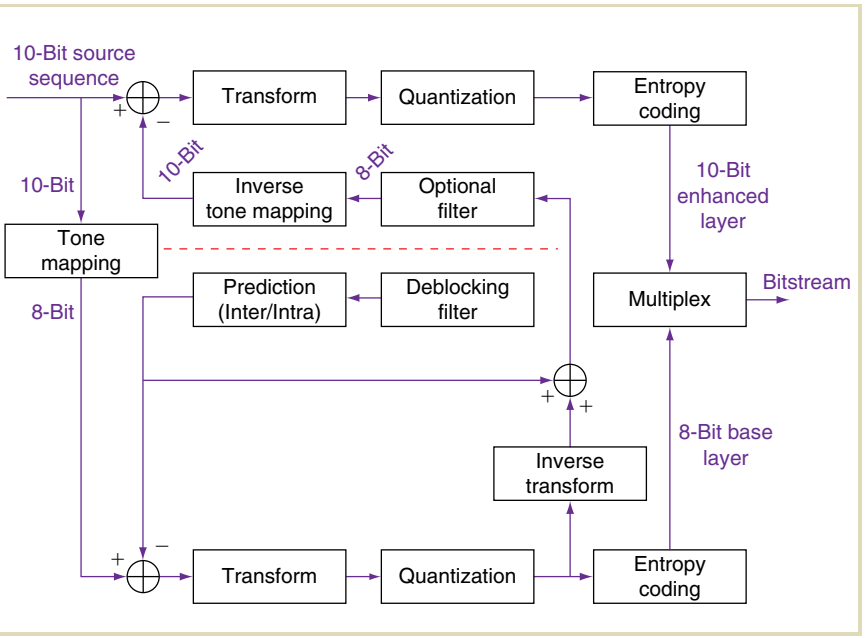


FIGURE 4.9 *Scalable bit-depth encoding in the H.264/AVC framework. (Redrawn from [366])*

decoder, and in this way, backwards-compatibility is achieved and full advantage of 8-bit compression efficiency at modern encoder architectures is taken.

For the recently emerging scalable H.264/AVC standard, the base layer should be compatible with its Main or High Profiles used in SDTV and HDTV, respectively. As in the backwards-compatible HDR MPEG-4 approach presented in the previous section, it is required that the 8-bit layer is encoded with original color grading, which cannot be modified. Also, as in HDR MPEG-4 encoding, an important issue is the *interlayer prediction* of the original higher-bit-depth (HDR) frame. This is computed using an inverse tone-mapping function, based on the decoded base layer. The residual difference between the high-bit-depth frame and the interlayer prediction is transformed, quantized, and stored in the enhancement layer. Finally, the bit streams from both layers are entropy coded and multiplexed into a scalable bit stream.

Note that in the encoding architecture, as presented in Figure 4.9, motion compensation is used for the base layer only, which after inverse tone mapping is then used in the interlayer prediction for the enhancement layer. The enhancement layer itself with the residual difference stream does not require any motion compensation, which greatly simplifies encoding and decoding [191].

The key issue for the overall compression efficiency is the design of the inverse tone-mapping module. Although the main goal of forward tone mapping is to produce LDR images of compelling visual quality, the only purpose of inverse tone mapping is to reduce the residual difference between the interlayer prediction and the original HDR video.

In particular, this means that it is not strictly required that inverse tone mapping is the inverse function of the forward tone-mapping curve. Moreover, the simplicity of inverse tone mapping means less information to transfer through the compressed HDR video stream to the decoder side, as well as reduced complexity of encoding and decoding itself. For this reason, Gao and Wu [100] used just a simple scaling of the base layer, with the scaling factor fixed for the whole sequence. However, such linear scaling performed poorly for highly nonlinear tone mapping.

Winken et al. [366] experimented with a global inverse tone mapping represented as a lookup table, which is optimized per frame (or a sequence of frames) by minimizing the mean square error between the inverse tone-mapped 8-bit samples and the original 10-bit samples. Essentially, this approach is similar to the MPEG-4

approach discussed in Section 4.2.1 [204], although it is embedded into the scalable H.264/AVC architecture.

This approach may lead to an excessive size of the enhancement layer for spatially varying (local) tone-mapping operators (see Figure 4.8 and Section 4.2.1). Clearly, localization of the inverse tone mapping that follows local changes in tone mapping is needed. To address this problem, Segal [289] proposed a linear inverse tone-mapping model with two parameters: the gain (α) and *offset*, which are both optimized for each macro-block to achieve the best prediction of its HDR pixels:

$$Y_{\mathrm{HDR}} = \alpha\, Y_{\mathrm{LDR}} + \mathit{offset}, \tag{4.7}$$

Here, Y_{HDR} and Y_{LDR} denote luma code words in the HDR and LDR frames. To reduce the storage cost and simplify decoding, the same α and *offset* are used for the chroma Cb_{HDR} and Cr_{HDR} prediction in the corresponding macro-block based on its LDR counterpart Cb_{LDR} and Cr_{LDR}. The problem with this approach is that video codecs typically transmit nonlinear (gamma-encoded) luma and chroma codewords instead of linear luminance and chrominance. The resulting nonlinear color spaces are not isoluminant, which means that luma information is present in the chroma components. To reduce the variance in chroma due to luma, the correction factors $Cb_{\mathrm{LDR,DC}}/Y_{\mathrm{LDR,DC}}$ and $Cr_{\mathrm{LDR,DC}}/Y_{\mathrm{LDR,DC}}$ have been introduced in the *offset* computation:

$$\begin{aligned} Cb_{\mathrm{HDR}} &= \alpha\, Cb_{\mathrm{LDR}} + \mathit{offset} \cdot \frac{Cb_{\mathrm{LDR,DC}}}{Y_{\mathrm{LDR,DC}}}, \\ Cr_{\mathrm{HDR}} &= \alpha\, Cr_{\mathrm{LDR}} + \mathit{offset} \cdot \frac{Cr_{\mathrm{LDR,DC}}}{Y_{\mathrm{LDR,DC}}}, \end{aligned} \tag{4.8}$$

where, $Y_{\mathrm{LDR,DC}}$, $Cb_{\mathrm{LDR,DC}}$, $Cr_{\mathrm{LDR,DC}}$ denote the DC portion (mean) of the luma and chroma components in the LDR image macro-block. Such DC values can be extracted from the DCT coefficients for each macro-block in the base layer.

Liu et al. [191] also propose a block-based interlayer prediction, but they differ with respect to Segal [289] as they keep the α and *offset* parameters for the luma and each of the chroma components independently. To reduce the cost of per-macro-block storage for all these extra parameters, they compare their values

with the top and left neighbor macro-blocks, and only the differences in these predictions are entropy coded and stored in the bit stream. In this way, Liu et al. avoid storing redundant gain and offset parameters for each macro-block in those image regions that exhibit only small changes in local tone mapping. Also, their interlayer prediction is more precise, as inverse tone mapping can be used independently per color component. This method results in a better compression than other methods discussed here. In particular, this is the case for HDR video sequences for which complex local tone mapping has been applied to the base layer.

In all methods discussed in this chapter, only one base and one enhancement layer is considered. However, it would be easy to extend scalable coding schemes, in particular those relying on linear scaling, to support more enhancement layers, such as 10-bit and 12-bit. Wu et al. [371] also experiment with combined bit depth and spatial scalability, where the image resolution is also enhanced.

HDR Image Capture

05

High dynamic range (HDR) images may be captured from real scenes or rendered using 3D computer graphics (CG) techniques such as radiosity and ray tracing. A few modern graphics cards are even capable of generating HDR images directly. The larger topic of CG rendering is well covered in other textbooks [95,293,177, 141,75]. In this chapter, the focus is on practical methods for capturing high-quality HDR images from real scenes using conventional camera equipment. In addition, commercial hardware designed to capture HDR images directly is beginning to enter the market, which is discussed toward the end of this chapter.

5.1 PHOTOGRAPHY AND LIGHT MEASUREMENT

A camera is essentially an imperfect device for measuring the radiance distribution of a scene, in that it cannot capture the full spectral content and dynamic range. (See Chapter 2 for definitions of "color" and "radiance.") The film or image sensor in a conventional or digital camera is exposed to the color and dynamic range of a scene, as the lens is a passive element that merely refocuses the incoming light onto the image plane. All the information is there, but limitations in sensor design prevent cameras from capturing all of it. Film cameras record a greater dynamic range than their digital counterparts, especially when they expose a negative emulsion.

© 2010, Elsevier Inc. All rights reserved.
DOI: 10.1016/B978-0-12-374914-7.00005-1

Standard black-and-white film emulsions have an inverse response to light, as do color negative films. Figure 5.1 shows example response curves for two film emulsions, demonstrating a sensitive range of nearly 4 log units or a 10 000:1 contrast ratio. Depending on the quality of the lens and the blackness of the camera's interior, some degree of "flare" may inhibit this range, particularly around a bright point such as the sun or its reflection. While the capability to record 4 log units of dynamic range is there, flare may reduce the effective dynamic range somewhat.

The film development process may also limit or enhance the information that is retrieved from the exposed emulsion, but the final constraining factor is of course the printing process. It is here where tone mapping takes place, since the effective

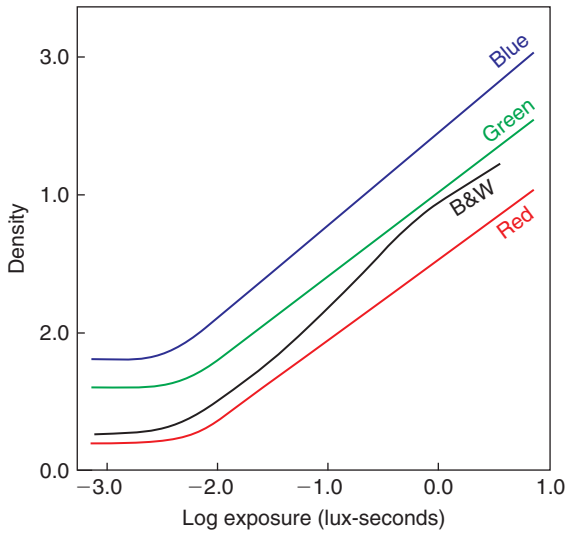


FIGURE 5.1 *Characteristic film curves showing response for color (R, G, and B) and black-and-white negative films over nearly four orders of magnitude.*

dynamic range of a black-and-white or color print is about 100:1 at best. Darkroom techniques such as dodge-and-burn may be used to get the most out of a negative, though in an industrial film-processing lab what usually happens is more akin to autoexposure after the fact.

To extract the full dynamic range from a negative, we need to digitize the developed negative or apply a "dry developing method" such as the one developed by Applied Science Fiction and marketed in the Kodak Film Processing Station [76]. Assuming that a developed negative is available, a film scanner would be required that records the full log range of the negative in an HDR format. Unfortunately, no such device exists, although it is technically feasible. However, one can take a standard film scanner, which records either into a 12-bit linear or an 8-bit sRGB color space, and use multiple exposures to obtain a medium dynamic range result from a single negative. The process is identical to the idealized case for multiple-exposure HDR capture, which we describe in the following section. The same method may be used to obtain an HDR image from a sequence of exposures using a standard digital camera or to enhance the dynamic range possible with a film camera.

5.1.1 CAMERA RAW

High-end Digital Single-Lens Reflex camera (DSLRs) are pushing the boundary of what can be captured in a single digital exposure, some recording as many as 14 bits per channel. Bear in mind that 14 bits in a linear encoding are not the same as 14 bits in a gamma or log encoding, as we learned in Chapter 3. Since the Charge-Coupled Device (CCD) and Complementary Metal-Oxide-Semiconductor (CMOS) sensors used in current DSLRs are inherently linear, so are the 14 bits of raw data they capture. Depending on ISO sensitivity and white balance, noise and clamping further reduce the useful range to less than the number of bits would indicate. In the end, the added precision is more useful for reducing banding at the top end than it is for increasing the dynamic range. This is because we are fundamentally limited by the capabilities of the capture device, which are in turn constrained by physical limitations of photo site size, thermal, and shot noise. While we have great hope that these limitations will ultimately be overcome, merely adding bits to the A/D converter without improving the underlying capture technology gets us only so far.

To illustrate this point, we took two series of photographs using a Nikon D700 camera, which has excellent noise performance thanks to its large, modest resolution sensor.[1] From this series, we assembled the HDR image shown tone-mapped in the top of Figure 5.2. Below the figure, we have blown up the region shown in white. On the left, we have the result from assembling the HDR from multiple exposures. In the middle, we enlarge the same area from a single ISO 200 exposure captured in 12-bit RAW mode, showing visible sensor noise. This noise is expected since we took the shortest exposure from our sequence in an attempt to capture the full range of the scene. We did not quite manage to include the sun's reflection, but we captured about three orders of magnitude above the noise floor, demonstrating the superior latitude of the D700's sensor. However, the visible difference in our results for 12-bit versus 14-bit RAW mode is almost negligible. Since the noise floor of the sensor is already greater than the quantization error of a 12-bit linear encoding, there is little benefit to increasing the number of bits; we only end up recording our noise more precisely. To get a truly HDR image, we still must rely on multiple exposures to bring the signal above the sensor noise while still capturing the brightest points in our scene.

5.2 HDR IMAGE CAPTURE FROM MULTIPLE EXPOSURES

Due to the limitations inherent in most digital image sensors, and to a lesser degree in film emulsions, it is not generally possible to capture the full dynamic range of a scene in a single exposure. However, by recording multiple exposures, a standard camera with the right software can create a single, HDR image (a.k.a. a *radiance map* as defined in Chapter 2). These exposures are usually captured by the camera itself, though in the case of recovering HDR information from a single negative, the same technique may be applied during the film-scanning phase.

By taking multiple exposures, each image in the sequence will have different pixels properly exposed and other pixels under- or overexposed. However, each pixel will be properly exposed in one or more images in the sequence. It is therefore

1 As resolution increases, so does noise due to the shrinking area of each pixel. For the best sensitivity and noise performance, larger, lower-resolution sensors are preferred.

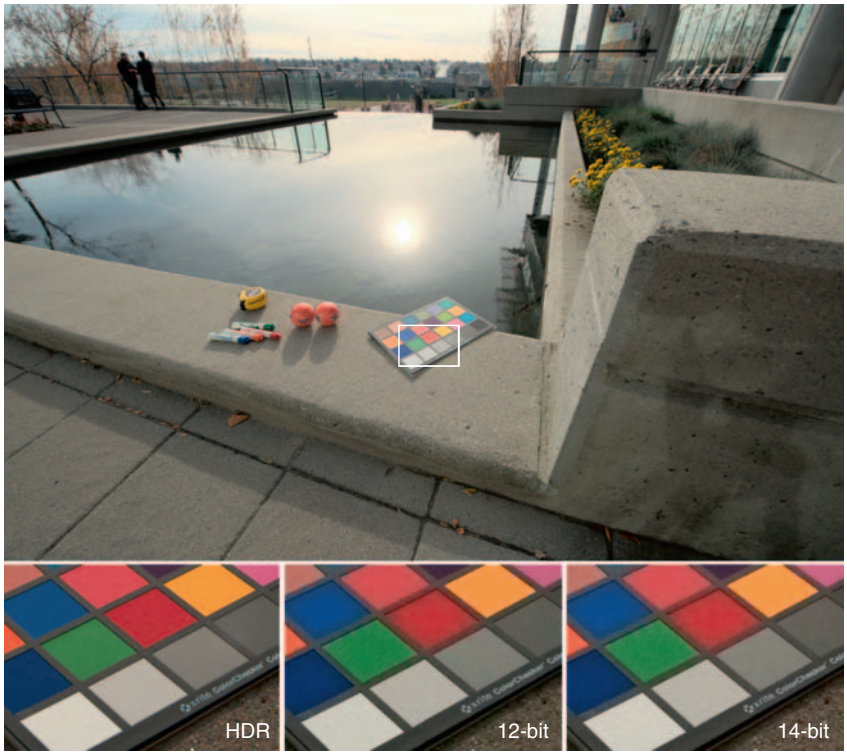


FIGURE 5.2 *An HDR scene captured in 12-bit and 14-bit RAW mode. The left enlargement shows a combined, noise-free HDR result. The middle enlargement shows the noise in a 12-bit RAW exposure that attempts to capture the entire range. The right enlargement shows similar noise in a 14-bit RAW capture. (Images courtesy Gerwin Damberg of Dolby Canada)*

possible and desirable to ignore very dark and very bright pixels in the following computations.

Under the assumption that the capturing device is perfectly linear, each exposure may be brought into the same domain by dividing each pixel by the image's exposure time. From the recorded radiance values L_e, this effectively recovers irradiance values E_e by factoring out the exposure duration.[2]

Once each image is in the same units of measurement, corresponding pixels may be averaged across exposures — of course excluding under- and overexposed pixels. The result is an HDR image. Assuming that weight factors are given by $w(Z_{ij})$ for 8-bit pixel values $Z_{ij} = L_e(i, j) = E_e(i, j)\,\Delta t_k$ at location (i, j), then an HDR image L_{ij} is computed from N exposures with

$$L_{ij} = \sum_{k=1}^{N} \frac{Z_{ij}\ w(Z_{ij})}{\Delta t_k} \Bigg/ \sum_{k=1}^{N} w(Z_{ij}) \tag{5.1}$$

Here, the exposure time is given by Δt_k for exposure k.

In practice, cameras are not perfectly linear light-measurement devices, so it becomes important to account for the camera's response. Section 5.7 describes methods for deriving such response functions. Assuming for now that the camera response $f()$ is known and given by $Z_{ij} = f(L_e(i, j)) = f(E_e(i, j)\,\Delta t_k)$, we can invert this function to recover exposures $E_e\,\Delta t_k$ from pixel values Z_{ij}. This leads to a new algorithm for constructing HDR images:

$$L_{ij} = \sum_{k=1}^{N} \frac{f^{-1}(Z_{ij})\ w(Z_{ij})}{\Delta t_k} \Bigg/ \sum_{k=1}^{N} w(Z_{ij}) \tag{5.2}$$

Further, objects frequently do not remain still between individual exposures, and the camera is rarely kept still. Thus, in practice, the above procedure needs to be refined to include image-alignment techniques, as well as ghost and lens flare removal.

2 The quantity captured by the camera is spectrally weighted radiance. As such, calling this quantity "radiance" is inappropriate. However, the spectral response curve usually differs from the CIE $V(\lambda)$ curve, and therefore this quantity also cannot be called "luminance" [62]. When the term "radiance" or "irradiance" is used, it should be understood that this refers to spectrally weighted radiance and irradiance.

Extracting a medium dynamic range radiance map from a single negative is relatively straightforward since it does not require alignment of multiple frames and does not suffer from object displacement that may occur during the capture of several exposures. It therefore serves as the basis for the techniques presented later in this chapter.

5.3 FILM SCANNING

In the ideal case for creating an HDR image from multiple low dynamic range (LDR) exposures, the scene or image should be completely static (e.g., an exposed and developed negative). We assume that the response curve of the film is known. In addition, the LDR capture device, such as an 8-bit/primary film scanner with known response curves, should provide some means to control the exposure exactly during multiple captures.

Creating an HDR image under these conditions starts by taking scans with multiple exposures. In addition, the system response is inverted to get back to a linear relation between scene radiances and pixel values. Each scanned image is multiplied by a calibration factor related to its exposure and combined into an HDR result. The only question is what weighting function to use in averaging together the linear exposures. Of course, the lightest and darkest pixels at the limits of each exposure should be excluded from consideration because these pixels are under- or overexposed. But how should the pixels in between be weighed?

Mann and Picard proposed a certainty/weighting function equal to the derivative of the system response curve for each color channel, using the argument that greater response sensitivity corresponds to greater certainty [201]. Debevec and Malik used a simple hat function based on the assumption that mid-range pixels are more reliable [62]. Mitsunaga and Nayar used signal theory to argue for multiplying Mann and Picard's weight by the response output, since larger values are less influenced by a constant noise floor [224]. Any of these methods will yield a satisfactory result, though the latter weighting function is better supported by signal theory. The Mitsunaga–Nayar weighting seems to work best when multiplied by a broad hat function, as shown in Figure 5.3. This avoids dubious pixels near the extremes, where gamut limitations and clamping may affect the output values unpredictably.

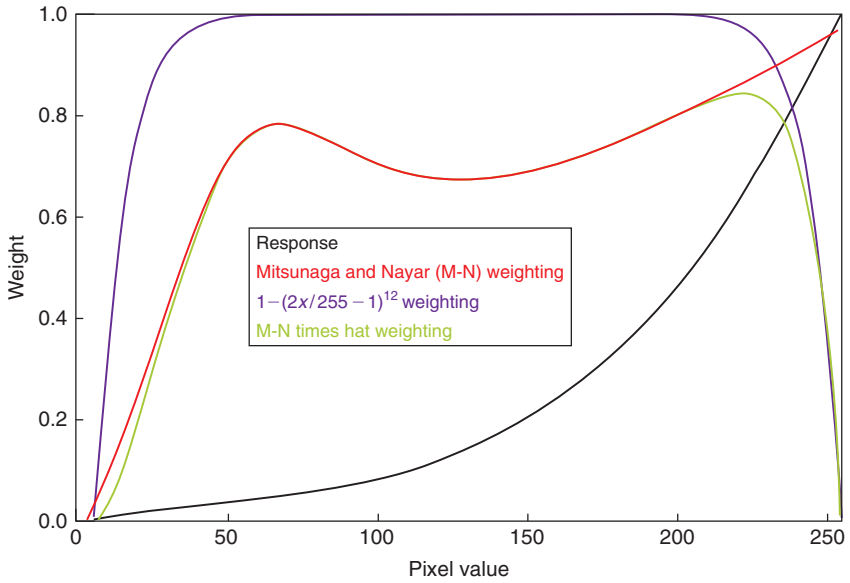


FIGURE 5.3 *The inverted system response function (solid line) and recommended Mitsunaga–Nayar weighting function, multiplied by an additional hat function* $1 - (2x/255 - 1)^{12}$ *to devalue the extrema, which are often suspect.*

Figure 5.4 shows a sequence of perfectly aligned exposures. Figure 5.5 (left image) shows the weighting used for each of the three contributing exposures, where blue is used for the longest exposure, green for the middle exposure, and red for the shortest exposure. As this figure shows, most pixels are a mixture of multiple exposures, with some pixels relying solely on the extremes of the exposure range. Figure 5.5 (right image) shows the combined result, tone mapped using a histogram adjustment operator [347].

FIGURE 5.4 *Our example exposure sequence. Each image is separated by two f-stops (equal to a factor of 4, or 0.6 $\log_{10}$ units).*

If the multiple exposures come not from multiple scans of a single negative but from multiple negatives or digital images, combining images may become problematic. First, the camera may shift slightly between exposures, which results in some subtle (and possibly not-so-subtle) misalignments that blur the results. Second, if the actual system response function is unknown, the images must be aligned before it can be estimated from the given exposures. Third, objects in the scene may shift slightly between frames or even make large movements, such as people walking in the scene as the photos are taken. Finally, flare in the camera lens may fog areas surrounding particularly bright image regions, which may not be noticeable in a standard LDR image. We address each of these problems in turn and present some work-arounds in the following sections.

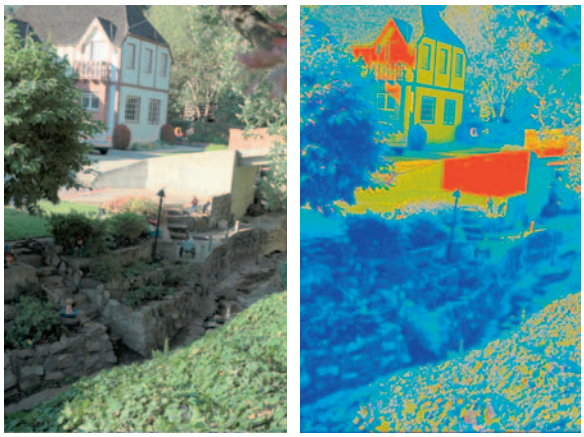


FIGURE 5.5 *The combined HDR result, tone mapped using the histogram adjustment operator (described in Section 8.5.1) in the right image. The left image shows contributing input image weights, where blue shows where the longer exposure dominates, green the middle, and red the shorter exposure. Most output pixels are an average of two or more input values, which reduce noise in the final result.*

5.4 IMAGE REGISTRATION/ALIGNMENT

Although several techniques have been developed or suggested for image alignment and registration, most originating from the computer vision community, only a few published techniques address the specific problem of aligning differently exposed frames for the purpose of HDR image creation. The first technique, by Kang et al. [147], handles both camera movement and object movement in a scene and is based on a variant of the Lucas and Kanade motion estimation technique [196]. In an off-line postprocessing step, for each pixel, a motion vector is computed between successive frames. This motion vector is then refined with additional

techniques such as hierarchical homography introduced by Kang et al. to handle degenerate cases.

Once the motion of each pixel is determined, neighboring frames are warped and thus registered with one another. Then, the images are ready to be combined into an HDR radiance map. The advantage of this technique is that it compensates for fairly significant motion and is for instance suitable for capturing HDR video by exposing successive frames by different amounts of time.

A second alignment technique, described below, uses a *median threshold bitmap* (MTB) [345]. This technique is also about 10 times faster than technique of Kang et al., since it performs its alignment operations on bitmaps rather than 8-bit grayscale images and does not perform image warping or resampling. However, the MTB alignment algorithm does not address moving objects in the scene and is not appropriate for arbitrary camera movements such as zooming and tilting. The method of Kang et al. may, therefore, be preferred in cases where arbitrary camera movement is expected. In the case of object motion, we recommend a simpler and more robust postprocessing technique in Section 5.9.

5.5 THE MEDIAN THRESHOLD BITMAP ALIGNMENT TECHNIQUE

In this section, we describe a method for the automatic alignment of HDR exposures [345].[3] Input to this exposure algorithm is a series of N 8-bit grayscale images, which may be approximated using only the green channel or derived from 24-bit sRGB with integer arithmetic[4]:

$$Y = (54\ R + 183\ G + 19\ B)/256$$

One of the N images is arbitrarily selected as the reference image, and the output of the algorithm is a series of $N - 1$ (x, y) integer offsets for each of the remaining

3 Reprinted with permission of A K Peters, Ltd. from Greg Ward, "Fast, Robust Image Registration for Compositing High Dynamic Range Photographs from Hand-Held Exposures," *Journal of Graphics Tools* Volume 8(2):17–30, 2003.

4 This is a close approximation of the computation of luminance as specified by ITU-R Rec. BT.709; its equation is given on p. 36.

images relative to this reference. These exposures may then be recombined efficiently into an HDR image using the camera response function, as described in Section 5.7.

The computation focuses on integer pixel offsets, because they can be used to quickly recombine the exposures without resampling. Empirical evidence suggests that handheld sequences do not require rotational alignment in approximately 90% of the cases. In a sequence where there is some discernible rotation, the modified method presented in Section 5.5.5 may be applied.

Conventional approaches to image alignment often fail when applied to images with large exposure variations. In particular, edge-detection filters are dependent on image exposure, as shown in the left side of Figure 5.6, where edges appear and disappear at different exposure levels. Edge-matching algorithms are therefore ill-suited to the exposure alignment problem when the camera response is unknown.

The MTB approach incorporates the following features:

- Alignment is done on bi-level images using fast bit-manipulation routines.
- The technique is insensitive to image exposure.
- For robustness, it includes noise filtering.

The results of a typical alignment are shown in Figure 5.12 in Section 5.5.4.

If we are to rely on operations such as moving, multiplying, and subtracting pixels over an entire high-resolution image, the algorithm is bound to be computationally expensive, unless our operations are very fast. Bitmap images allow us to operate on 32 or 64 pixels at a time using bitwise integer operations, which are very fast compared with bytewise arithmetic. We use a bitmap representation that facilitates image alignment independent of exposure level, the MTB. The MTB is defined as follows:

1. Determine the median 8-bit value from a low-resolution histogram over the grayscale image pixels.
2. Create a bitmap image with 0's where the input pixels are less than or equal to the median value and 1's where the pixels are greater.

Figure 5.6 shows two exposures of an Italian stairwell in the middle and their corresponding edge maps on the left and MTBs on the right. In contrast to the

FIGURE 5.6 *Two unaligned exposures (middle) and their corresponding edge bitmaps (left) and median threshold bitmaps (right). The edge bitmaps are not used precisely because of their tendency to shift dramatically from one exposure level to another. In contrast, the MTB is stable with respect to exposure.*

edge maps, the MTBs are nearly identical for the two exposures. Taking the difference of these two bitmaps with an exclusive-or (XOR) operator shows where the two images are misaligned, and small adjustments in the x and y offsets yield predictable changes in this difference due to object coherence (e.g., the left image in Figure 5.10). However, this is not the case for the edge maps, which are noticeably different for the two exposures, even though we attempted to compensate for the camera's nonlinearity with an approximate response curve. Taking the difference of the two edge bitmaps would not give a good indication of where the edges are misaligned, and small changes in the x and y offsets yield unpredictable results, making gradient search problematic. More sophisticated methods to determine edge correspondence are necessary to use this information, and we can avoid these and their associated computational costs with the MTB-based technique.

The constancy of an MTB with respect to exposure is a very desirable property for determining image alignment. For most HDR reconstruction algorithms, the alignment step must be completed before the camera response can be determined, since the response function is derived from corresponding pixels in the different exposures. By its nature, an MTB is the same for any exposure within the usable range of the camera, regardless of the response curve. So long as the camera's response function is monotonic with respect to world radiance, the same scene will theoretically produce the same MTB at any exposure level. This is because the MTB partitions the pixels into two equal populations, one brighter and one darker than the scene's median value. Since the median value does not change in a static scene, the derived bitmaps likewise do not change with exposure level.[5]

There may be certain exposure pairs that are either too light or too dark to use the median value as a threshold without suffering from noise, and for these, we choose either the 17th or the 83rd percentile as the threshold, respectively. Although the offset results are all relative to a designated reference exposure, we actually compute offsets between adjacent exposures, so the same threshold may be applied to both images. Choosing percentiles other than the 50th (median) results in fewer pixels to compare, and this makes the solution less stable, so we may choose to limit the maximum offset in certain cases. The behavior of percentile threshold bitmaps is

5 Technically, the median value could change with changing boundaries as the camera moves, but such small changes in the median are usually swamped by noise, which is removed by this algorithm, as we will explain.

otherwise the same as the MTB, including stability over different exposures. In the remainder of this section, when we refer to the properties and operations of MTBs, the same applies for other percentile threshold bitmaps as well.

Once the threshold bitmaps corresponding to the two exposures have been computed, there are several ways to align them. One brute-force approach is to test every offset within the allowed range, computing the XOR difference at each offset and taking the coordinate pair corresponding to the minimum difference. A more efficient approach might follow a gradient descent to a local minimum, computing only local bitmaps differences between the starting offset (0,0) and the nearest minimum. We prefer a third method based on an image pyramid that is as fast as gradient descent in most cases but is more likely to find the global minimum within the allowed offset range.

Multiscale techniques are well-known in the computer vision and image-processing communities, and image pyramids are frequently used for registration and alignment (see, e.g., [321]). This technique starts by computing an image pyramid for each grayscale image exposure, with $\log_2$(max_offset) levels past the base resolution. The resulting MTBs are shown for two example exposures in Figure 5.7. For each smaller level in the pyramid, we take the previous grayscale image and filter it down by a factor of two in each dimension, computing the MTB from the grayscale result. The bitmaps themselves should not be subsampled, as the result will be subtly different and could potentially cause the algorithm to fail.

To compute the overall offset for alignment, we start with the lowest-resolution MTB pair and compute the minimum difference offset between them within a range of ± 1 pixel in each dimension. At the next resolution level, this offset is multiplied by 2 (corresponding to the change in resolution), and we compute the minimum difference offset within a ± 1 pixel range of this previous offset. This continues to the highest (original) resolution MTB, where we get the final offset result. Thus, each level in the pyramid corresponds to a binary bit in the computed offset value.

At each level, we need to compare exactly nine candidate MTB offsets, and the cost of this comparison is proportional to the size of the bitmaps. The total time required for alignment is thus linear with respect to the original image resolution and independent of the maximum offset, since the registration step is linear in the number of pixels, and the additional pixels in an image pyramid are determined by the size of the source image and the (fixed) height of the pyramid.

FIGURE 5.7 *A pyramid of MTBs is used to align adjacent exposures one bit at a time. The smallest (rightmost) image pair corresponds to the most significant bit in the final offset.*

5.5.1 THRESHOLD NOISE

The algorithm just described works well in images that have a fairly bimodal brightness distribution but can run into trouble for exposures that have a large number of pixels near the median value. In such cases, the noise in near-median pixels shows up as noise in the MTB, which destabilizes the difference computations.

The inset in Figure 5.8 shows a close-up of the pixels in the dark stairwell exposure MTB, which is representative for the type of noise seen in some images. Computing the XOR difference between exposures with large areas like these yields noisy results that are unstable with respect to translation, because the pixels

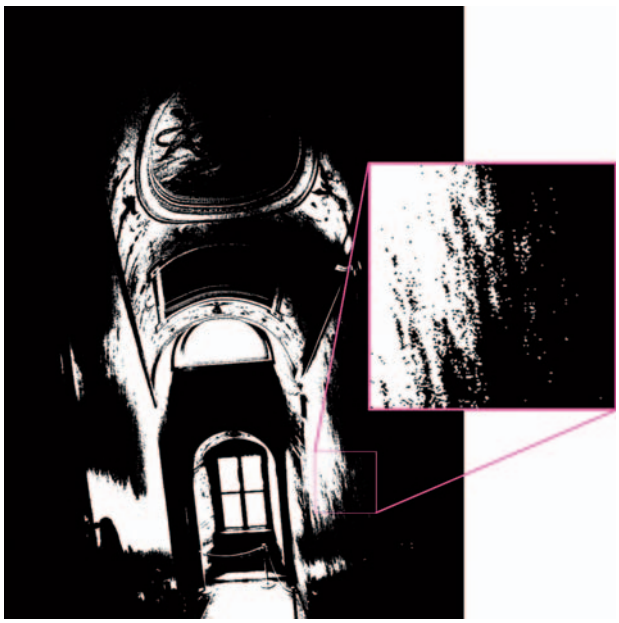


FIGURE 5.8 *Close-up detail of noisy area of MTB in dark stairwell exposure (full resolution).*

themselves tend to move around in different exposures. Fortunately, there is a straightforward solution to this problem.

Since this problem involves pixels whose values are close to the threshold, these pixels can be excluded from our difference calculation with an *exclusion bitmap*. The exclusion bitmap consists of 0's wherever the grayscale value is within some specified distance of the threshold and 1's elsewhere. The exclusion bitmap for the exposure in Figure 5.8 is shown in Figure 5.9, where all bits are zeroed where pixels are within ± 4 of the median value.

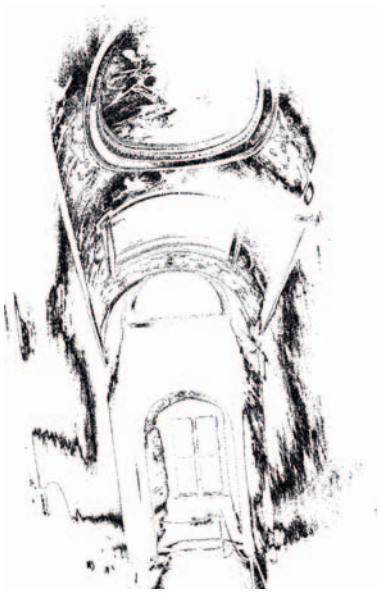


FIGURE 5.9 *An exclusion bitmap, with zeroes (black) wherever pixels in our original image are within the noise tolerance of the median value.*

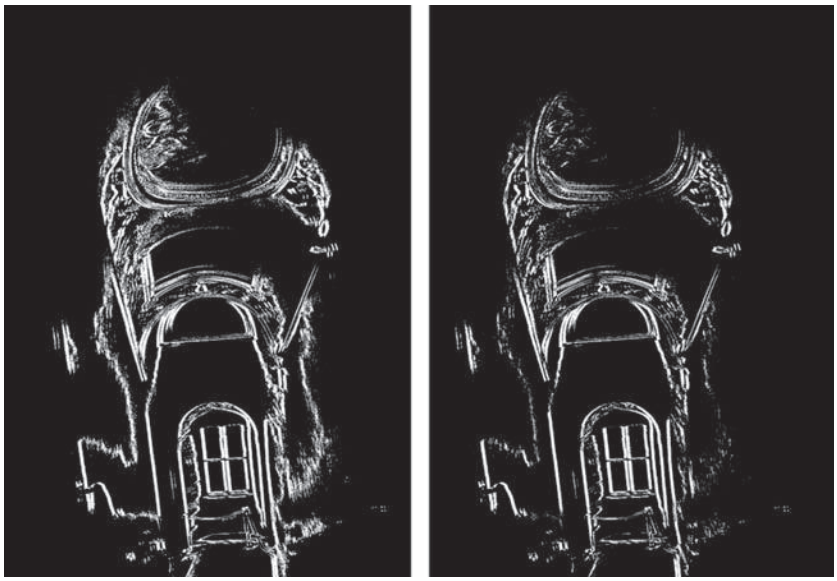


FIGURE 5.10 *The original XOR difference of the unaligned exposures (left) and with the two exclusion bitmaps ANDed into the result to reduce noise in the comparison (right).*

We compute an exclusion bitmap for each exposure at each resolution level in the MTB pyramid, then take the XOR difference result for each candidate offset, ANDing it with both offset exclusion bitmaps to compute the final difference.[6] The effect is to disregard differences that are less than the noise tolerance in our images. This is illustrated in Figure 5.10, where the XOR difference of the unaligned exposures before and after applying the exclusion bitmaps is shown. By removing

6 If we were to AND the exclusion bitmaps with the original MTBs before the XOR operation, we would inadvertently count disagreements about what was noise and what was not as actual pixel differences.

those pixels that are close to the median, the least reliable bit positions in the smooth gradients are cleared, but the high confidence pixels near strong boundaries, such as the edges of the window and doorway, are preserved. Empirically, this optimization seems to be very effective in eliminating false minima in the offset search algorithm.

5.5.2 OVERALL ALGORITHM

The full algorithm with the exclusion operator is given in the recursive C function, `GetExpShift`, shown in Figure 5.11. This function takes two exposure images and determines how much to move the second exposure (`img2`) in x and y to align

```
GetExpShift (const Image *img1, const Image *img2,
             int shift_bits, int shift_ret[2])
{
  int    min_err;
  int    cur_shift[2];
  Bitmap tb1, tb2;
  Bitmap eb1, eb2;
  int    i, j;
  if (shift_bits > 0) {
    Image sml_img1, sml_img2;
    ImageShrink2(img1, &sml_img1);
    ImageShrink2(img2, &sml_img2);
    GetExpShift(&sml_img1, &sml_img2, shift_bits-1, cur_shift);
    ImageFree(&sml_img1);
    ImageFree(&sml_img2);
    cur_shift[0] *= 2;
    cur_shift[1] *= 2;
  } else
    cur_shift[0] = cur_shift[1] = 0;
```

FIGURE 5.11 *The* `GetExpShift` *algorithm.*

```
ComputeBitmaps(img1, &tb1, &eb1);
ComputeBitmaps(img2, &tb2, &eb2);
min_err = img1->xres * img1->yres;
for (i = -1; i <= 1; i++)
  for (j = -1; j <= 1; j++) {
    int     xs = cur_shift[0] + i;
    int     ys = cur_shift[1] + j;
    Bitmap  shifted_tb2;
    Bitmap  shifted_eb2;
    Bitmap  diff_b;
    int     err;
    BitmapNew(img1->xres, img1->yres, &shifted_tb2);
    BitmapNew(img1->xres, img1->yres, &shifted_eb2);
    BitmapNew(img1->xres, img1->yres, &diff_b);
    BitmapShift(&tb2, xs, ys, &shifted_tb2);
    BitmapShift(&eb2, xs, ys, &shifted_eb2);
    BitmapXOR(&tb1, &shifted_tb2, &diff_b);
    BitmapAND(&diff_b, &eb1, &diff_b);
    BitmapAND(&diff_b, &shifted_eb2, &diff_b);
    err = BitmapTotal(&diff_b);
    if (err < min_err) {
      shift_ret[0] = xs;
      shift_ret[1] = ys;
      min_err = err;
    }
    BitmapFree(&shifted_tb2);
    BitmapFree(&shifted_eb2);
  }
  BitmapFree(&tb1); BitmapFree(&eb1);
  BitmapFree(&tb2); BitmapFree(&eb2);
}
```

FIGURE 5.11 (continued)

it with the first exposure (`img1`). The maximum number of bits in the final offsets is determined by the `shift_bits` parameter. The more important functions called by `GetExpShift` are

`ImageShrink2 (const Image *img, Image *img_ret)` Subsample the image `img` by a factor of two in each dimension and put the result into a newly allocated image `img_ret`.

`ComputeBitmaps (const Image *img, Bitmap *tb, Bitmap *eb)` Allocate and compute the threshold bitmap `tb` and the exclusion bitmap `eb` for the image `img`. (The threshold and tolerance to use are included in the image struct.)

`BitmapShift (const Bitmap *bm, int xo, int yo, Bitmap *bm_ret)` Shift a bitmap by (`xo`,`yo`) and put the result into the preallocated bitmap `bm_ret`, clearing exposed border areas to zero.

`BitmapXOR (const Bitmap *bm1, const Bitmap *bm2, Bitmap *bm_ret)` Compute the "XOR" of `bm1` and `bm2` and put the result into `bm_ret`.

`BitmapTotal (const Bitmap *bm)` Compute the sum of all 1 bits in the bitmap.

Computing the alignment offset between two adjacent exposures is simply a matter of calling the `GetExpShift` routine with the two image structs (`img1` and `img2`), which contain their respective threshold and tolerance values. (The threshold values must correspond to the same population percentiles in the two exposures.) We also specify the maximum number of bits allowed in the returned offset, `shift_bits`. The shift results computed and returned in `shift_ret` will thus be restricted to a range of $\pm\, 2^{\texttt{shift\_bits}}$.

There is only one subtle point in the above algorithm, which is what happens at the image boundaries. Unless proper care is taken, nonzero bits may inadvertently be shifted into the candidate image. These would then be counted as differences in the two exposures, which would be a mistake. It is therefore crucial that the `BitmapShift` function shifts 0's into the new image areas so that applying

the shifted exclusion bitmap to the XOR difference will clear these exposed edge pixels as well. This also explains why the maximum shift offset needs to be limited. In the case of an unbounded maximum shift offset, the lowest difference solution will also have the least pixels in common between the two exposures — one exposure will end up shifted completely off the other. In practice, we have found a `shift_bits` limit of 6 (±64 pixels) to work fairly well.

5.5.3 EFFICIENCY CONSIDERATIONS

Clearly, the efficiency of the MTB alignment algorithm depends on the efficiency of the bitmap operations, as 9 shift tests with 6 whole-image bitmap operations apiece are performed. The `BitmapXOR` and `BitmapAND` operations are easy enough to implement, as we simply apply bitwise operations on 32-bit or 64-bit words, but the `BitmapShift` and `BitmapTotal` operators may not be so obvious.

For the `BitmapShift` operator, any two-dimensional shift in a bitmap image can be reduced to a one-dimensional shift in the underlying bits, accompanied by a clear operation on one or two edges for the exposed borders. Implementing a one-dimensional shift of a bit array requires at most a left or right shift of B bits per word with a reassignment of the underlying word positions. Clearing the borders then requires clearing words where sequences of 32 or 64 bits are contiguous and partial clears of the remainder words. The overall cost of this operator, although greater than the XOR or AND operators, is still modest. This `BitmapShift` implementation includes an additional Boolean parameter that turns off border clearing. This optimizes the shifting of the threshold bitmaps, which have their borders cleared later by the exclusion bitmap, so that the `BitmapShift` operator does not need to clear them.

For the `BitmapTotal` operator, a table of 256 integers is computed corresponding to the number of 1 bits in the binary values from 0 to 255 (i.e., 0, 1, 1, 2, 1, 2, 2, 3, 1, . . . , 8). Each word of the bitmap can then be broken into byte-sized chunks and used to look up the corresponding bit counts from the precomputed table. The bit counts are then summed together to yield the correct total. This results in a speedup of at least eight times over counting individual bits and may be further accelerated by special-case checking for zero words, which occur frequently in this application.

5.5.4 RESULTS

Figure 5.12 shows the results of applying the MTB image-alignment algorithm to all five exposures of the Italian stairwell, with detailed close-ups showing before and after alignment. The misalignment shown is typical of a handheld exposure sequence, requiring translation of several pixels on average to bring the exposures back atop each other. We have found that even tripod exposures sometimes need minor adjustments of a few pixels for optimal results.

After applying this translational alignment algorithm to over 100 handheld exposure sequences, a success rate of approximately 84% was found, with 10% giving unsatisfactory results due to image rotation. About 3% failed due to excessive scene motion, usually waves or ripples on water that happened to be near the threshold value and moved between frames, and another 3% had too much high-frequency

FIGURE 5.12 *An HDR image composited from unaligned exposures (left) and detail (top center). Exposures aligned with the MTB algorithm yield a superior composite (right) with clear details (bottom center).*

content, which made the MTB correspondences unstable. Most of the rotation failures were mild, leaving at least a portion of the HDR image well-aligned. Other failures were more dramatic, throwing alignment off to the point where it was better not to apply any translation at all.

5.5.5 ADDING ROTATIONAL ALIGNMENT

It is not too difficult to add rotational alignment to the basic MTB technique. An approximate technique aligns balanced image sections. Figure 5.13 shows three paired alignment regions used to derive an average rotation angle for a particular

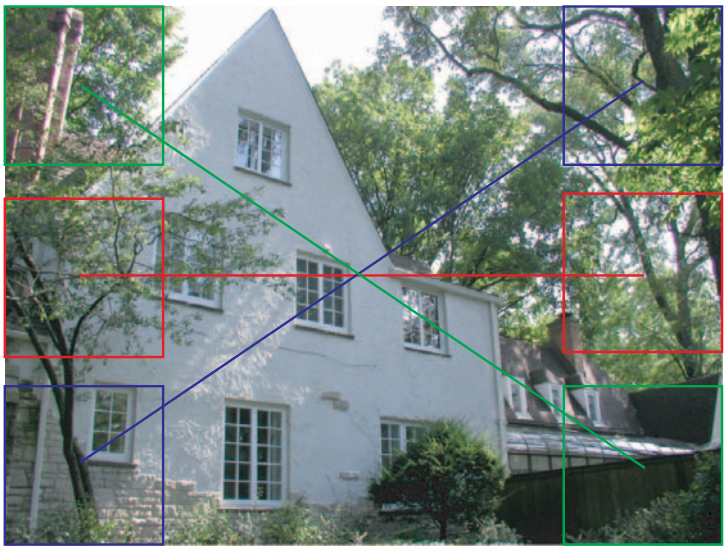


FIGURE 5.13 *Three paired, color-coded regions are used to determine rotational alignment between exposures using improved MTB technique.*

FIGURE 5.14 *The left image (top inset) shows the original MTB translational alignment on an HDR capture sequence with significant rotation. The right image (bottom inset) shows the same sequence rotationally aligned with our modified MTB method.*

image sequence. Each region in the exposure is aligned with the adjacent exposure using the MTB technique. The three pairs of translational values are then averaged into an overall rotation to apply between exposures. After this rotation, a final translational alignment is computed over the entire image, just as before. This technique works well for modest rotations up to several degrees, after which our subregion alignments lose reliability. Figure 5.14 compares pure translational alignment with our paired-region rotation plus translation alignment method.

5.6 OTHER ALIGNMENT METHODS

A more robust alignment technique also based on the MTB was developed by Thorsten Grosch [115]. Grosch computes MTB alignment many times in a simplex optimization of rotation and translation using graphics hardware. Jacobs et al.

used a similar approach a year earlier [139], but without the benefit of hardware acceleration, so their convergence took several minutes. Both papers also discussed methods for reducing the appearance of "ghosts" caused by scene motion, which we discuss in Section 5.9.

5.7 DERIVING THE CAMERA RESPONSE FUNCTION

Combining LDR exposures into an HDR image requires knowledge of the camera response function to linearize the data. In general, the response function is not provided by camera makers, who consider it part of their proprietary product differentiation. Assuming that an sRGB response curve (as described in Chapter 2) is unwise, as most makers boost image contrast beyond the standard sRGB gamma to produce a livelier image. There is often some modification as well at the ends of the curves, to provide softer highlights and reduce noise visibility in the shadows. However, as long as the response is not altered by the camera from one exposure to the next, it is possible to deduce this function given a proper image sequence.

5.7.1 DEBEVEC AND MALIK TECHNIQUE

Debevec and Malik [62] demonstrated a simple and robust technique for deriving the camera response function from a series of aligned exposures, extending earlier work by Mann and Picard [201]. The essential idea is that by capturing different exposures of a static scene, one is effectively sampling the camera response function at each pixel. This is best demonstrated graphically.

Figure 5.15 shows three separate image positions sampled at five different exposures shown in Figure 5.16. The relative exposure ratios at each of the three positions are given by the speed settings on the camera; thus, we know the shape of the response function at three different parts of the curve. However, we do not know how these three curve fragments fit together. Debevec and Malik resolved this problem using linear optimization to find a smooth curve that minimizes the mean squared error over the derived response function. The objective function they use

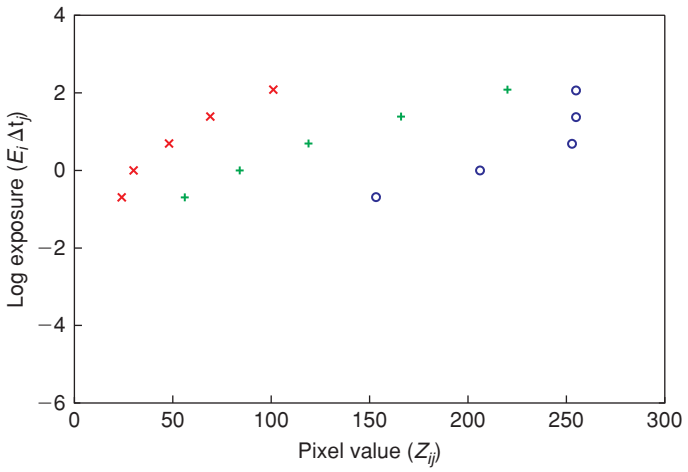


FIGURE 5.15 Plot of $g(Z_{ij})$ from three pixels observed in five images, assuming unit radiance at each pixel. The images are shown in Figure 5.16.

is to derive the logarithmic inverse response function $g(Z_{ij})$:

$$\mathcal{O} = \sum_{i=1}^{N} \sum_{j=1}^{P} \left\{ w(Z_{ij}) \left[g(Z_{ij}) - \ln E_i - \ln \Delta t_j \right] \right\}^2 + \lambda \sum_{z=Z_{\min}+1}^{Z_{\max}-1} \left[w(z)\, g''(z) \right]^2,$$

where Δt_j is the exposure time for exposure j, E_i is the film irradiance value at image position i, and Z_{ij} is the recorded pixel at position i and exposure j.

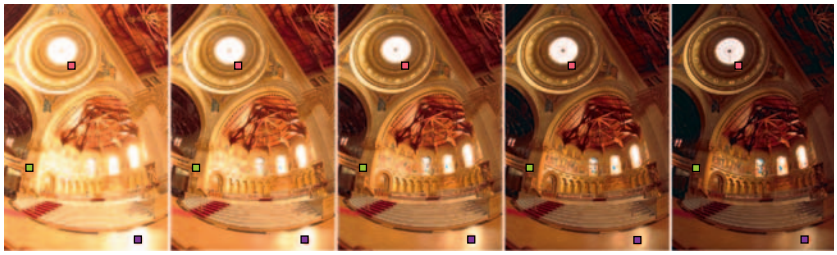


FIGURE 5.16 *Three sample positions over five exposures shown in Figure 5.15.*

The logarithmic response function $g(Z_{ij})$ relates to the camera response function as follows:

$$g(Z_{ij}) = \ln f^{-1}(Z_{ij})$$

The weighting function $w(Z_{ij})$ is a simple hat function:

$$w(z) = \begin{cases} z - Z_{\min} & \text{for } z \leq \frac{1}{2}(Z_{\min} + Z + \max) \\ Z_{\max} - z & \text{for } z > \frac{1}{2}(Z_{\min} + Z_{\max}) \end{cases}$$

This equation is solved using singular value decomposition to obtain an optimal value for $g(Z)$ for every possible value of Z or 0–255 for an 8-bit image. Each of the RGB channels is treated separately, yielding three independent response functions. This assumes that interactions between the channels can be neglected. Although this assumption is difficult to defend from what we know about camera color transformations, it seems to work fairly well in practice.

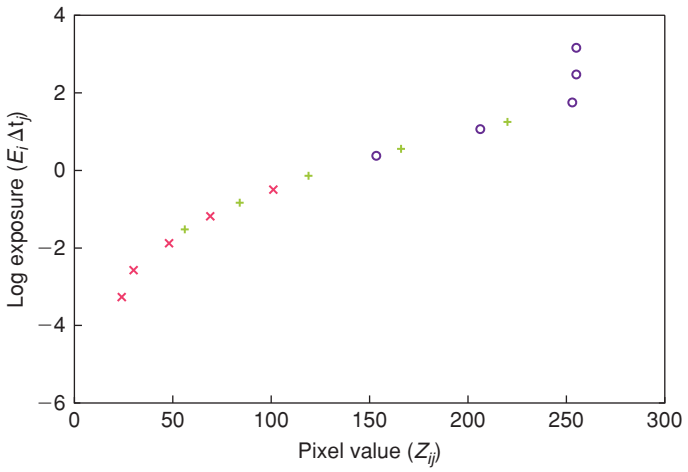


FIGURE 5.17 *Normalized plot of $g(Z_{ij})$ after determining pixel exposures.*

Figure 5.17 shows the result of aligning the three curves from Figure 5.15, and by applying the minimization technique to many image samples, it is possible to obtain a smooth response function for each channel.

5.7.2 MITSUNAGA AND NAYAR TECHNIQUE

Mitsunaga and Nayar presented a similar approach where they derive a polynomial approximation to the response function [224] rather than the enumerated table of Debevec and Malik. The chief advantage they cite in their technique is the ability to resolve the exact exposure ratios in addition to the camera response function. This proves important for lower-cost consumer equipment whose aperture and shutter speed may not be known exactly.

Mitsunaga and Nayar define the following N-dimensional polynomial for their camera response function

$$f(M) = \sum_{n=0}^{N} c_n M^n$$

Note: For consistency with Debevec and Malik, we suggest the following variable replacements:

$$M \rightarrow Z$$
$$N \rightarrow K$$
$$n \rightarrow k$$
$$Q \rightarrow P$$
$$q \rightarrow j$$

The final response function is thus defined by the $N + 1$ coefficients of this polynomial, $\{c_0, \ldots c_N\}$. To determine these coefficients, they minimize the following error function for a given candidate exposure ratio, $R_{q,q+1}$ (the scale ratio between exposure q and $q + 1$):

$$\varepsilon = \sum_{q=1}^{Q-1} \sum_{p=1}^{P} \left[\sum_{n=0}^{N} c_n M_{p,q}^n - R_{q,q+1} \sum_{n=0}^{N} c_n M_{p,q+1}^n \right]^2$$

The minimum is found by determining where the partial derivatives with respect to the polynomial coefficients are all zero, that is, solving the following system of $N + 1$ linear equations:

$$\frac{\partial \varepsilon}{\partial c_n} = 0$$

As in previous methods, they only solve for the response up to some arbitrary scaling. By defining $f(1) = 1$, they reduce the dimensionality of their linear system by

one coefficient, substituting:

$$c_N = 1 - \sum_{n=0}^{N-1} c_n$$

The final $N \times N$ system can be written:

$$\begin{bmatrix} \sum_{q=1}^{Q-1}\sum_{p=1}^{P} d_{p,q,0}\left(d_{p,q,0} - d_{p,q,N}\right) & \dots & \sum_{q=1}^{Q-1}\sum_{p=1}^{P} d_{p,q,0}\left(d_{p,q,N-1} - d_{p,q,N}\right) \\ \dots & \dots & \dots \\ \sum_{q=1}^{Q-1}\sum_{p=1}^{P} d_{p,q,N-1}\left(d_{p,q,0} - d_{p,q,N}\right) & \dots & \sum_{q=1}^{Q-1}\sum_{p=1}^{P} d_{p,q,N-1}\left(d_{p,q,N-1} - d_{p,q,N}\right) \end{bmatrix}$$

$$\times \begin{bmatrix} c_0 \\ \dots \\ c_{N-1} \end{bmatrix} = \begin{bmatrix} -\sum_{q=1}^{Q-1}\sum_{p=1}^{P} d_{p,q,0}\, d_{p,q,N} \\ \dots \\ -\sum_{q=1}^{Q-1}\sum_{p=1}^{P} d_{p,q,N-1}\, d_{p,q,N} \end{bmatrix}$$

where

$$d_{p,q,n} = M^n_{p,q} - R_{q,q+1}\, M^n_{p,q+1}$$

The original Mitsunaga and Nayar formulation only considers adjacent exposures. In practice, the system is more stable if all exposure combinations are considered. The error function becomes a triple sum by including a sum over $q' \neq q$ instead of just comparing q to $q+1$. This then gets repeated in the sums of the combined system of equations, where $d_{p,q,n}$ is replaced by

$$d_{p,q,q',n} = M^n_{p,q} - R_{q,q'}\, M^n_{p,q'}$$

To compute the actual exposure ratios between images, Mitsunaga and Nayar apply an interactive technique, where the system of equations above is solved repeatedly,

and between each solution, the exposure ratios are updated using

$$R_{q,q+1}^{(k)} = \sum_{p=1}^{P} \frac{\sum_{n=0}^{N} c_n^{(k)} M_{p,q}^n}{\sum_{n=0}^{N} c_n^{(k)} M_{p,q+1}^n}$$

Iteration is complete when the polynomial is no longer changing significantly:

$$|f_{(k)}^{-1}(M) - f_{(k-1)}^{-1}(M)| < \varepsilon, \qquad \forall\, M$$

This leaves just one final problem—what is the polynomial degree, N? The authors recommend solving for every degree polynomial up to some maximum exponent (e.g., 10), accepting the solution with the smallest error, ε. Fortunately, the solution process proceeds quickly, so this is not much of a burden. It is a good idea to ensure that the same degree is selected for all color channels, so a combined ε function is preferable for this final test.

Figure 5.18 shows the recovered response function for this sequence fitted with a third-order polynomial using Mitsunaga and Nayar's method and compares it to the standard sRGB response function. The camera produces an artificially exaggerated contrast with deeper blacks on its LDR exposures. This sort of response manipulation is fairly standard for consumer-grade cameras and many professional SLRs as well.

5.7.3 CHOOSING IMAGE SAMPLES FOR RESPONSE RECOVERY

Each of the techniques described for camera response recovery requires a set of intelligently selected samples from the exposure sequence. In principle, one could use every pixel from every image, but this would only add to the computation time while actually reducing stability in the solution due to misaligned and noisy data. Once the exposures have been aligned, the following procedure for selecting sample patches is recommended:

1. Sort the exposures from lightest to darkest.
2. Select an appropriate sample patch size and an optimal number of patches, and initialize (clear) the patch list.
3. Determine how many patches from the previous exposure are still valid for this one.

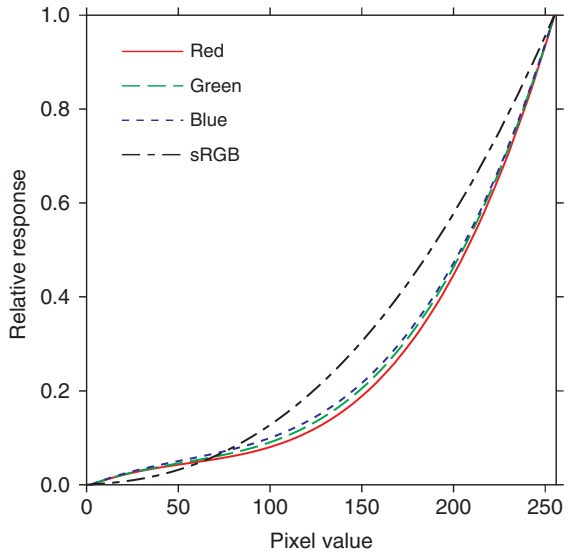


FIGURE 5.18 *Recovered red, green, and blue response functions for the image sequence shown in Figure 5.19.*

4 Compute how many more patches we need for this exposure — if none, then go to next exposure (Step 3).
5 Search for valid patches using randomized rejection sampling — a valid patch is brighter than any of the previous exposure's patches, does not overlap any other patch, and possesses a low internal variance. It is also within the valid range for this exposure.
6 Once we have found enough patches or given up due to an excess of rejected samples, we continue to the next exposure (Step 3).

A target of fifty 12×12 pixel patches per exposure seems to work well. In cases where the darker exposures do not use their full range, it becomes difficult to find new patches that are brighter than the previous exposure. In practice, this does not affect the result significantly, but it is important for this reason to place a limit on the rejection-sampling process in Step 5, lest we go into an infinite loop.

Figure 5.19 shows an exposure sequence and the corresponding patch locations. Adjacent exposures have nearly the same patch samples, but no patch sample survives in all exposures. This is due to the range restriction applied in Step 5 to

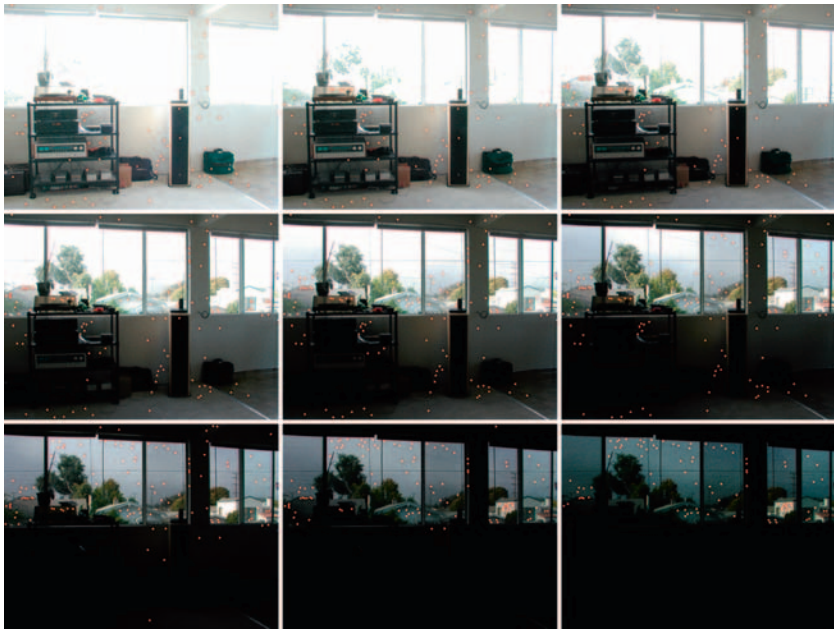


FIGURE 5.19 *Red squares indicate size and location of patch samples in each exposure.*

avoid unreliable pixel values. Figure 5.20 shows a close-up of the middle exposure with the patches shown as boxes, demonstrating the low variance in the selected regions. By rejecting high-contrast areas, errors due to exposure misalignment and sensor noise are minimized.

FIGURE 5.20 *Close-up of central exposure, showing selected patch regions.*

5.7.4 DERIVING SAMPLE VALUES FROM HISTOGRAMS

Another way to choose samples avoids the issue of pixel correspondence entirely by working instead on the image histograms. In 2002, Grossberg and Nayar introduced the idea of matching histogram percentiles rather than image positions to derive the camera response function in the presence of modest scene and camera motion [116]. Figure 5.21 shows how these correspondences are made. As we saw in the MTB alignment technique described in Section 5.4, percentiles relate to actual values in the scene, and so long as we can correlate the same values from one exposure to the next, it does not really matter where they exist in each exposure. At any given

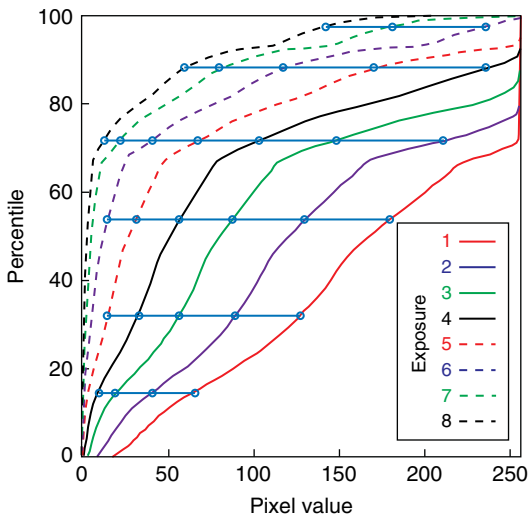


FIGURE 5.21 *A set of cumulative histograms for eight exposures and the percentile correspondences between them. Rather than matching patches in the underlying exposures, histogram matching avoids issues of scene and camera motion for camera response recovery.*

percentile value, we can find corresponding values in the cumulative histograms of adjacent exposures, which takes the place of the corresponding pixel regions found earlier. These points on our response functions may likewise be plugged into the Mitsunaga–Nayar technique, the Debevec–Malik method, or Grossberg and Nayar's direct derivation described in the same 2002 paper.

5.7.5 CAVEATS AND CALIBRATION

To apply these techniques successfully, it helps to follow some additional guidelines:

- Use aperture priority or manual exposure mode so that only the exposure time is allowed to vary. This reduces problems associated with vignetting (light fall-off toward the edge of the image).
- Fix the camera's white balance on a specific setting for the entire sequence — preferably daylight (a.k.a. D_{65}).
- If the camera offers an "optimized color and contrast" mode, switch it off. The more settings you can fix manually, the less likely the camera will alter the response function between exposures. This especially goes for automatic ISO/ASA and programmed exposure modes.
- Use a tripod if possible, and control your camera via a tether to a laptop computer if this option is available. The less touching of the camera during a sequence, the fewer alignment problems you will experience.

In general, it works best to calibrate your camera's response one time, then reuse this calibration for later exposure sequences. In this way, the scene and exposure sequence may be optimized for camera response recovery. For such a sequence:

- Set the camera on a tripod and use a tether if available. Alignment may still be necessary between exposures if the camera is touched during the sequence, but the method described in the previous section works far better with a tripod than a with a handheld sequence where the exposure is being changed manually.
- Choose a scene with large, gray or white surfaces that provide continuous gradients for sampling. The closer your scene is to a neutral color, the less likely color transforms in the camera will undermine the response recovery process.

- Choose a scene with very bright and very dark areas, then take a long sequence of exposures separated by 1 EV (a factor of two in exposure time). The darkest exposure should have no RGB values greater than 200 or so, and the lightest exposure should have no RGB values less than 20 or so. Do not include an excess of exposures beyond this range, as it will do nothing to help with response recovery and may hurt.
- If you have access to an luminance meter, take a reading on a gray card or uniform area in your scene to provide absolute response calibration.

Once a camera has been characterized in this way, it is possible to combine handheld bracketed sequences that are too short to reliably recover the response function.

5.8 NOISE REMOVAL

Cameras are designed to work optimally under a somewhat limited range of settings. For conventional photography, the exposure time, aperture, and ISO sensitivity contribute to the appearance of the image, trading the time it takes to capture the shot against depth of field, amount of light capture, and amount of noise present. Normally, the photographer chooses settings both creatively and for the purpose of minimizing artifacts.

In multiexposure photography, we deliberately under- and overexpose our images to capture as much of the scene as possible, in both light and dark regions. While this works in theory, and reasonably well in practice, there is a good chance that noise occurs, especially in very short exposures.

The subsequent processing means that this noise is further amplified. This can be seen by inspecting Equation 5.2, where after weighting and linearizing, pixels are divided by exposure time Δt_k. This means that pixel values in short exposures become emphasized relative to pixel values in long exposures. However, it is precisely the short exposures where we may expect noise.

It is therefore reasonable to assume that noise may play an adverse role in HDR photography. A solution to this problem can be found by noting that summing exposures as in Equation 5.2 is a special case of *frame averaging* [25]. The latter technique takes several noisy but otherwise identical images and averages them pixel by pixel.

We can borrow from frame averaging and modify the basic HDR image composition equation without taking any further exposures [9]. The idea is to apply frame averaging as a preprocess to condition each exposure. For exposure k, we use exposures $k+1, \ldots, k+s$, where $s < N-k$. The pixel values Z_{ij} are replaced with pixel values Z'_{ij} that exhibit less noise. The equation is akin to the basic multiple exposure summing method (Equation 5.2, albeit that the weight function $w()$ is replaced with w_{nr}. This leads to the following set of equations:

$$Z'_{ijk} = f(\Delta t_k \, c_k) \tag{5.3}$$

$$c_k = \sum_{p=k}^{k+m} \frac{f^{-1}(Z_{ijp}) \; w_f(Z_{ijp}, k)}{\Delta t_p} \Bigg/ \sum_{p=k}^{k+m} w_f(Z_{ijp}, k) \tag{5.4}$$

$$w_f(Z_{ijp}, k) = \begin{cases} \Delta t_k & p = k \\ \tau(Z_{ijk}) \; \Delta t_k & p \neq k \end{cases} \tag{5.5}$$

where $f()$ is the recovered camera response function. The new pixel values Z'_{ijk} at pixel position (i, j) in exposure k have reduced noise characteristics and can now be used in a standard HDR image-assembly process (Equation 5.2). The weight function w_f consists of two components, namely $\tau(Z_{ijk})$ and t_k. The former is used to give progressively less weight to pixel values that are close to being overexposed. It is given by

$$\tau(x) = \begin{cases} 1 & 0 \leq x < 200 \\ 1 - 3h(x)^2 + 2h(x)^3 & 200 \leq x < 250 \\ 0 & 250 \leq x \leq 255 \end{cases} \tag{5.6}$$

where

$$h(x) = 1 - \frac{250 - x}{50} \tag{5.7}$$

Choosing the weight function τ to be essentially equal to the exposure time Δt_k exploits the fact that noise that can be removed from an image in this manner is photon shot noise, which is characterized by a Poisson distribution, that is, a distribution with a variance equal to the mean. This method is not suitable for removing other types of noise from an image, including dark current and read noise.

FIGURE 5.22 *The image on the left was assembled without removing noise, which is present due to the choice of a high ISO setting on the camera. The image on the right was created using the same exposures, but this time applying the noise-removal algorithm.*

An example result is shown in Figure 5.22. This figure shows that significant amounts of noise can be removed without adverse effects using this technique. This allows higher ISO settings on the camera and thus captures image sequences with shorter exposure times. The advantage of this is that other artifacts such as the occurrence of ghosts can be minimized, especially in low-light conditions.

5.9 GHOST REMOVAL

Once the exposures are aligned to each other and the camera's response curve is determined, we may safely combine the images as described in Section 5.3. However, if some person or object was moving during the image sequence acquisition, they may appear as "ghosts" in the combined result due to their multiple locations. The technique described by Kang et al. [147] attempts to address this problem during alignment by warping pixels according to local content, but even if this can be done correctly in the presence of people who change posture and position, it still

leaves the problem of filling in holes that were obstructed in some views but not in others.

Khan et al. [7] presented an iterative solution to the ghosting problem, where weights are adjusted in each contributing exposure based on an estimate of their probability of belonging to the "background" (i.e., static objects). This process does an excellent job of removing ghosts in scenes when many exposures have been captured, since the separation of foreground and background is more reliable the more exposures one has as input.

A simpler approach is based on the observation that each exposure in the sequence is self-consistent, so we can simply choose one exposure or another in specific regions to obtain a ghost-free result. In this process, we select a single *reference exposure* from our captured sequence. This is typically the image with the greatest number of in-range pixels, though it may also be taken as the image whose ghost regions are best kept in range [115].

Starting from such a reference exposure, Gallo et al. [99] use a gradient domain approach to avoid seams at the ghost boundaries, solving a Poisson equation locally and balancing color in each region. Good results may also be obtained with the simpler method of adjusting pixel weights based on deviation from the reference image. The following formula may be used to downgrade a pixel with value p that differs significantly from our reference r:

$$w = \frac{[a(r_0)]^2}{[a(r_0)]^2 + [(p-r)/r]^2}$$

Figure 5.24 graphs this function for a constant a of 0.058, which is appropriate only if our reference pixel is optimally exposed and our acceptable error is low. In fact, a needs to vary depending on where we are in the reference image, which is why we make this a function of r_0, the unconverted LDR reference normalized to a 0–1 range. We recommend the following function for a:

$$a(r_0) = \begin{cases} 0.058 + (r_0 - 0.85) * 0.68 & \text{for } r_0 \geq 0.85 \\ 0.04 + (1 - r_0) * 0.12 & \text{for } r_0 < 0.85 \end{cases}$$

Where the reference pixel r_0 is too light or too dark to be useful, we leave the original weights as they were.

FIGURE 5.23 *Five exposures combined into a single HDR image, where people moving through the scene have caused ghosting in the result.*

Adjusting weights in this way biases image regions that are valid in the reference exposure toward that one point in time. Other exposure regions that are consistent with the reference image will be averaged in, whereas regions in other exposures that disagree (and might therefore cause ghosting) are downgraded. Pixels in the reference exposure that are too bright or too dark to be used will look as they did before without ghost removal. Thus, it is important to select as our reference the image that is well-exposed in the regions where we want ghosts removed.

Figure 5.23 shows an HDR image captured from a bracketed sequence of five exposures, excerpted in the left-hand side of the figure. People walking in and out of the temple result in a trail of ghosts appearing in the combined result.

Figure 5.25 shows the combined result with ghosts removed, where the longest exposure was used as the reference.

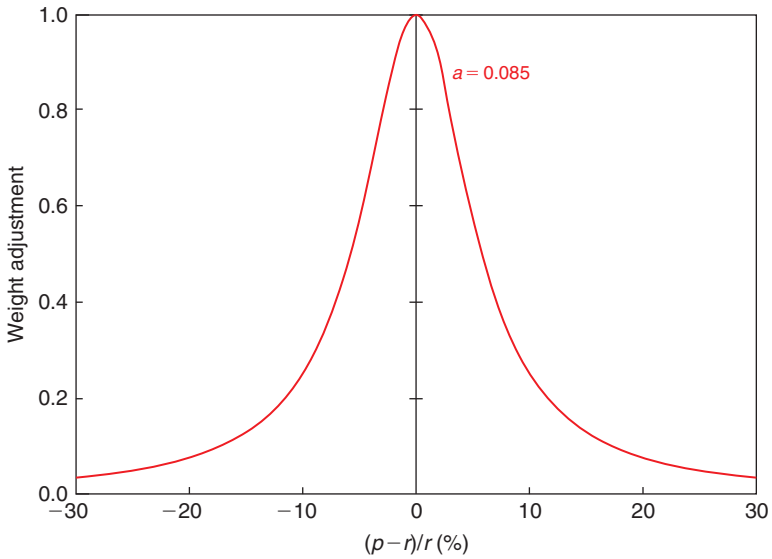


FIGURE 5.24 *The weight adjustment function reduces the influence of values that disagree with the reference exposure.*

5.10 LENS FLARE REMOVAL

After eliminating motion between exposures, there may still be artifacts present due to the camera's optics. Most digital cameras are equipped with optics that are consistent with the inherent limitations of 24-bit digital images. In other words, manufacturers generally do not expect more than two orders of magnitude to be captured in the final image, so certain parameters may be relaxed in the lens and sensor design relative to a 35-mm film camera. For an HDR capture process, however, the limitations of the system's optics are more apparent, even in a well-made

FIGURE 5.25 *The combined HDR result with ghosts removed.*

digital camera. Small issues such as the thickness and finish of the aperture vanes can make a big difference to the distribution of light on the sensor. The quality of coatings on the lenses and the darkness and geometry of the interior surrounding the sensor also come into play. Overall, there are many components that affect the scattering of light in an image, and it is difficult or impossible to arrive at a single set of measurements that characterize the system and its dependencies. Therefore, we prefer a dynamic solution to the lens flare problem, based only on the captured image.

Since it is important to keep all the optical properties of the camera consistent between exposures, normally only the shutter speed should be manipulated between exposures. Thus, the actual distribution of light on the sensor plane never varies, only the length of time the sensor is exposed to it. Therefore, any flare effects that are present in one exposure are present to the same degree in all exposures and will sum consistently into our HDR result. For this reason, there is no need and no sense to work on individual exposures, as it would only serve to increase our computational burden. The camera's *point spread function* (PSF) is a physical measure of the system optics, and it may be characterized directly from the recorded radiances in an HDR image.

5.10.1 THE POINT SPREAD FUNCTION

The PSF as it is defined here is an idealized, radially symmetric characterization of the light fall-off surrounding a point of light in a perfectly dark surrounding. It could be measured by making a pinhole in a piece of aluminum foil in front of a light bulb in a box and photographing it in a completely dark environment, as shown in Figure 5.26. The edge of the hole ought to be perfectly sharp, but it generally is not. The spread of light around the hole corresponds to light scattered within the lens of the digital camera.

This photograph of the pinhole could then be used to correct the combined HDR result for other photographs made with precisely the same lens settings — zoom and aperture. However, this procedure is much to expect of even the most meticulous photographer, and since lens flare also depends strongly on dust and oils that come and go over time, it is not practical to maintain a set of calibrated PSFs for any but the most critical applications.

However, there is a technique whereby the PSF may be approximated based on image content, as we demonstrate with the HDR capture shown in Figure 5.27. Despite the fact that this image contains no localized bright spots, and hence no easily measured PSF, a reasonable estimate of lens flare may still be obtained.

It is assumed that in the image, there exist some dark pixels near very bright (or "hot") pixels. To the extent this is true, it will be possible to estimate the PSF

FIGURE 5.26 *An isolated spot of light in a darkened environment for the purpose of measuring the point spread function of a camera.*

for the camera.[7] It is also assumed that the lens flare is radially symmetric. This is admittedly a crude approximation but required by the estimation procedure. Thus, the goal is to find and remove the radially symmetric component of flare. Streaks and other asymmetrical artifacts generated by the camera optics will remain.

The automatic flare removal consists of the following steps:

1. Compute two reduced-resolution HDR images, one in color and one in gray scale. Call these I_{CR} and I_{GR}, respectively.
2. Identify "hot" pixels in I_{GR}, which are over some threshold.

7 If this assumption is false, and there are no dark pixels near sources, then lens flare will probably go unnoticed and there is no need to remove it.

FIGURE 5.27 *Our input test image for estimating lens flare using the same aperture and zoom as in Figure 5.26 (tone mapped with the histogram equalization operator, described in Section 8.5.1).*

3. Drawing annuli around each hot pixel, compute a least squares approximation to the PSF using the method described in the following section.
4. Apply the PSF to remove flare from the final HDR image.

Reducing the working resolution of our HDR image achieves a major speedup without significantly impacting the quality of the results, since flare tends to be a distributed phenomenon. A reduced image size of at most 128 pixels horizontally

or vertically is sufficient. The threshold setting for Step 2 is not particularly important, but we have found a value of 1000 times the minimum (reduced) pixel value to work well for most images. Of course, a different threshold is advisable if the minimum is zero. Steps 3 and 4 require some explanation, which we give in the following subsections.

5.10.2 ESTIMATING THE PSF

The PSF defines how light falls off around bright points in the image.[8] To estimate the PSF, the minimum pixel values around all "hot" pixels in the image are measured, thus arriving at a conservative estimate of the PSF. To do this, the potential contributions of all hot pixels at a certain distance from the darker (non-"hot") pixels are summed to build up an estimate of the PSF from the corresponding minima, radius by radius. For example, Figure 5.28 shows an image with exactly three of these super-bright pixels. The same radius is drawn around all three pixels, creating three overlapping circles. If the PSF were known a priori, we could compute the contribution of these hot pixels at this distance by multiplying the PSF, which is a function of radius, by each hot pixel value. Conversely, dividing the darker pixels around each circle by the circle's center gives an upper bound to the PSF.

Furthermore, the PSF at this distance cannot be greater than the minimum of all darker pixel/hot pixel ratios. Where the circles overlap, the sum of hot pixel contributions should be considered. In fact, a convenient approach is to sum together all three hot pixel values around each circle in a separate image. The PSF upper bound at that radius will then equal the minimum ratio of the darker pixels in the circles to their hot pixel sums, the point marked with an "X" in the example.

This technique extends directly to any number of hot pixels. The estimation procedure follows:

1. For each radius we wish to consider:
   a. Sum hot pixel values into radius range (annulus) in a separate image.
   b. Find the minimum ratio of darker pixel/hot pixel sum for all annuli.

8 In fact, it defines how light falls off around any point in the image, but only the bright points matter since the fall-off is so dramatic.

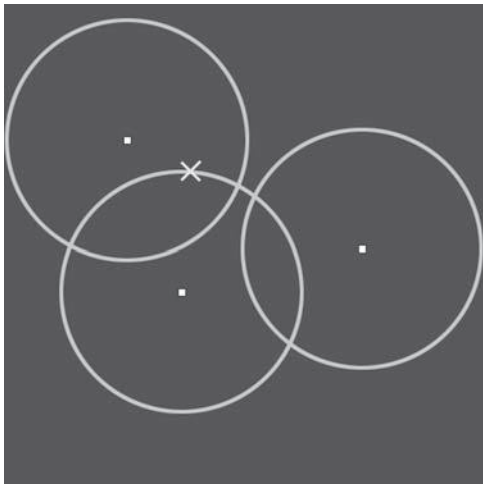


FIGURE 5.28 *An example image with exactly three bright pixels, surrounded by circles showing where their PSF influences overlap. Each PSF radius contains exactly one minimum, marked with an "X" in this example.*

2 If the minimum ratio is not less than the previous (smaller) radius, then discard it, since we assume that the PSF is monotonically decreasing.
3 For each minimum ratio pixel, identified for each sample radius, consider all flare contributions to this pixel over the entire image.

Once we have an estimate of the upper limit to the PSF at each radius, these minimum pixels can be used to fit a third-degree polynomial, $p(x)$, using the reciprocal of the input radius for x.[9] For each identified minimum pixel position with value P_i,

9 The use of a third-degree polynomial, and fitting to the reciprocal of distance, are heuristic choices we have found to produce good results at an economical cost.

we can write the following equation:

$$P_i = \sum_j P_j \left(C_0 + \frac{C_1}{r_{ij}} + \frac{C_2}{r_{ij}^2} + \frac{C_3}{r_{ij}^3} \right)$$

where the P_j's are the hot pixel values and r_{ij} are the distances between the minimum pixel P_i and each hot pixel position. This equation can be rewritten as

$$P_i = C_0 \sum_j P_j + C_1 \sum_j \frac{P_j}{r_{ij}} + C_2 \sum_j \frac{P_j}{r_{ij}^2} + C_3 \sum_j \frac{P_j}{r_{ij}^3}$$

The sums in the above equation then become coefficients in a linear system where the four fitting parameters (C_0 through C_3) are the unknowns. As long as there are more than four minimum pixel values, P_i, it should be possible to solve this as an over-determined system using standard least squares minimization. Heuristically, a better solution may be obtained if we assign minimum and maximum permitted values for the distance between pixels, r_{ij}. Anytime the actual distance is less than the minimum radius, 3 pixels in the reduced image, we use a distance of 3 instead. Similarly, the distance is clamped to a maximum of half the image width. This avoids stability problems and sensitivity to local features in our image. It also avoids the possibly incorrect removal of flare too close to light sources. This is generally impossible anyway, since flare from the lens can be so great that the underlying information is washed out. In such cases, no recovery can be made. This often happens at bright source boundaries.

Figure 5.29 compares the PSF measured in Figure 5.26 to our estimate derived solely from the input image in Figure 5.27. Other than the artificial plateau we imposed by constraining the minimum r_{ij} to 3 pixels, the two curves are a reasonable match. The fitted function shows a slightly greater flare than the measured one, but this is explained by the fact that the measurement was based on a spot near the center of the image. Optical flare becomes more pronounced as one moves farther toward the edges of an image, especially in a wide-angle lens. Since the fitting function was applied over the entire image, we would expect the globally estimated PSF to be slightly greater than a PSF measured at the center.

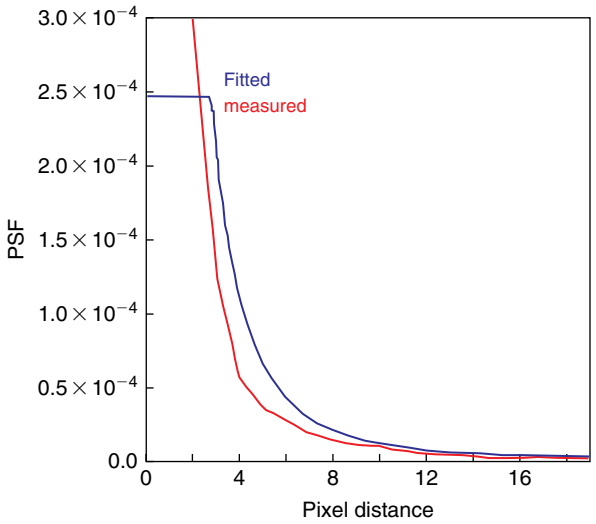


FIGURE 5.29 *Comparison between directly measured PSF from Figure 5.26 and function fitted using image in Figure 5.27.*

5.10.3 REMOVING THE PSF

Given an estimate of the PSF, flare removal is straightforward. For each hot pixel in the image, we subtract the PSF times this pixel value from its surroundings. Since the neighborhood under consideration may extend all the way to the edge of the image, this can be an expensive operation. Once again, working with a reduced image lowers the computational cost to manageable levels. The steps for removing the PSF are

1 Create a reduced-resolution flare image, $\mathbf{F}_{\text{CR}}$, and initialize it to black.

FIGURE 5.30 *An image of a rosette window, before flare removal (left) and after (center). The right image shows the estimated PSF applied to hot pixels in the image.*

2 For each hot pixel in the reduced image $\mathbf{I}_{CR}$, multiply by the PSF and add the product into $\mathbf{F}_{CR}$.
3 If the value of any pixel in $\mathbf{F}_{CR}$ is larger than its corresponding pixel in $\mathbf{I}_{CR}$, reduce the magnitude of $\mathbf{F}_{CR}$ uniformly to compensate.
4 Upsample $\mathbf{F}_{CR}$ using linear interpolation and subtract from the original HDR image.

Step 3 above ensures that no negative pixels are generated in the output and is necessary because the fitting method does not guarantee the most conservative PSF. Dependent on the interpolation and the local variance of the original pixels, we may still end up with negative values during Step 4 and should truncate these where they occur. An example result of automatic flare removal is shown in Figure 5.30, along with the reduced-resolution flare image generated during Step 2, above.

5.11 HDR CAPTURE HARDWARE

With the possible exception of lens flare removal, the techniques explained in the last section might be unnecessary if we had a digital sensor that could record the full dynamic range in a single shot. In fact, such sensors are being actively developed, and some are even being marketed, but only a few integrated solutions are commercially available. We describe some of these systems briefly in this section. However, as capture with multiple exposures is still the least expensive way to create HDR images, we begin this section by discussing practical techniques to capture multiple exposures.

5.11.1 CAPTURE OF MULTIPLE EXPOSURES

Cameras have various mechanisms to regulate the amount of light captured by the sensor. This includes setting the aperture, exposure time, and ISO value. It is also possible to use neutral density filters. However, the latter approach means that filters have to be exchanged during the capture process, which is slow and prone to cause misaligned and ghosting artifacts.

Changing the ISO value to higher settings creates more noise than necessary, while adjusting the aperture not only changes the amount of light incident upon the sensor but also depth of field. This is undesirable as this means that the exposures in the sequence are not all identical.

The best way to create HDR images by capturing multiple exposures is therefore to vary the exposure time. While it is possible to alter the exposure time manually, it is often more convenient to use the autobracketing feature found in most modern cameras. This captures the sequence as fast as the camera can manage, thereby minimizing the impact of camera and object movement.

The choice of camera is somewhat important here, as the maximum number of bracketed exposures that can be taken with a single button press depends strongly on the make and model. In general, more exposures means that a higher dynamic range can be captured in one go. However, most scenes can be effectively captured with nine exposures, whereas many more are within reach with a camera that allows 5–7 exposures to be bracketed. Table 5.1 shows a nonexhaustive list of cameras and their bracketed capabilities.

Make	Model	Autobracketed Frames	Maximum EV	Total f-stops
Canon	1D MKII / MKII N	7	3	18
Canon	1D MKIII	7	3	18
Canon	1Ds MKII	5	3	6
Canon	1Ds MKIII	7	3	18
Fuji	S5 Pro	9	1	8
Kodak	DCS Pro SLR/c	3	3	6
Nikon	D200	9	1	8
Nikon	D300	9	1	8
Nikon	D300s	9	1	8
Nikon	D700	9	1	8
Nikon	D2H	9	1	8
Nikon	D2X/D2Xs	9	1	8
Nikon	D3	9	1	8
Pentax	K7	5	2	8
Pentax	K10D	5	2	8
Pentax	K20D	5	2	8
Samsung	GX-10	5	2	8
Samsung	GX-20	5	2	8
Sigma	SD14	3	3	6
Sigma	DP1	3	3	6

TABLE 5.1 *Current still camera models capable of capturing 6 f-stops of dynamic range using autobracketing (after http://www.hdr-photography.com/aeb.html).*

5.11.2 VIPER FILMSTREAM

Grass Valley, a division of Thomson, introduced the Viper FilmStream^{tm} camera for digital cinematography in the fall of 2002 (http://www.thomsongrassvalley.com/products/cameras/viper/). This is currently the top-end performer for digital capture, and it produces an enormous amount of data, up to 444 MB/s! The camera contains three HDTV 1080i (1920 × 1080 resolution) CCD sensors, one each for red, green, and blue, and records directly into a 10-bit/channel log format. The camera and its response functions are shown in Figures 5.31 and 5.32.

This chart shows that the Viper captures about three orders of magnitude, which is at least 10 times what a standard digital video camera captures, and begins to rival film. The equipment is currently available for lease only from Thompson.

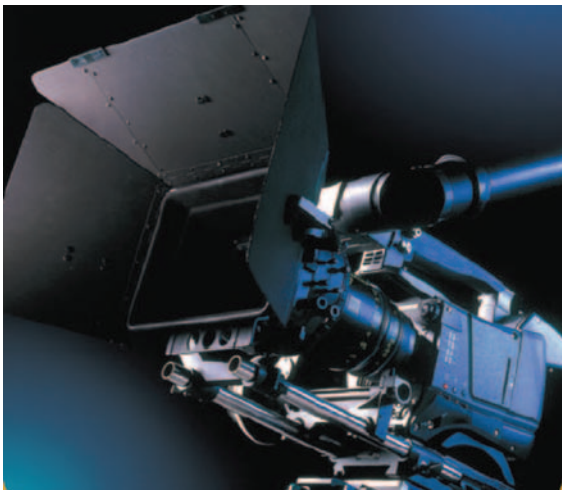


FIGURE 5.31 *The Viper FilmStream^{tm} camera.*

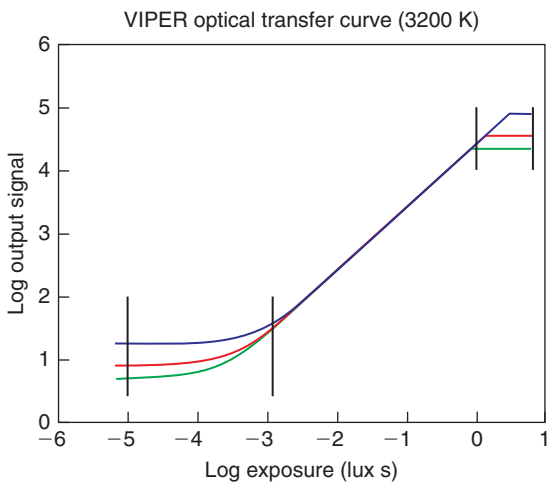


FIGURE 5.32 *The Viper FilmStream^{tm} camera response curve.*

5.11.3 PANAVISION

Panavision markets the Genesis, which is a digital film camera that outputs 10 bits per channel in log format, and can be regarded as a medium dynamic range camera. The sensor contains 12.4 Mpixels capturing a wide color gamut at up to 50 frames per second.

5.11.4 PIXIM

Pixim Inc. of Mountain View, California (http://www.pixim.com), offers two 720 × 480 CMOS image sensors that boast a 10-bit digital video output with over 100 dB signal-to-noise ratio, corresponding to roughly four orders of magnitude. These sensors grew out of the Programmable Digital Camera Project headed by

Abbas El Gamal and Brian Wandell at Stanford University and use "multisampling" on picture elements to minimize noise and saturation. Pixels are grouped with independent analog-to-digital converters and sampled multiple times during each video frame. Conversion at each sensor group stops when either the frame time is up or the value nears saturation. In effect, each pixel group has its own electronic shutter and dynamic exposure system. For additional processing, the sensor chip is paired with a custom digital image processor, which handles video conversion and control. Pixim currently markets the sensors and development kits to OEMs, which has been picked up by a few security camera makers, including Dallmeier, Sunell, and FSAN.

5.11.5 SPHERONVR

SpheronVR, AG of Kaiserslautern, Germany (http://www.spheron.com), has a very high-resolution and high-performance HDR camera on the market, the SpheroCam HDR. This device boasts the ability to capture full spherical panoramas at a resolution of up to 13,000×5300 pixels, covering nearly eight orders of magnitude in dynamic range (a 10^8:1 contrast ratio). However, since they use a line-scan CCD for their capture, the process takes "several minutes" to complete a full 360° scan at this resolution. Lower resolutions and dynamic ranges will scan more rapidly, but one can never achieve a single-shot capture with a line-scan camera, as the device must mechanically pan over the scene for its exposure. Nevertheless, this is the system to beat for panoramic capture and critical image-based lighting applications, and their deluxe package comes with an advanced software suite as well. Figure 5.33 shows a SpheroCam HDR image captured in Napa Valley, California, at a resolution of about 3000 × 2100. The dynamic range is 5.5 orders of magnitude.

5.11.6 WEISS AG

Weiss AG, also of Kaiserslautern, Germany (http://www.weiss-ag.com), markets an omnidirectional HDR still camera, named the Civetta. This camera is claimed to capture 28 f-stops of dynamic range. A spherical image of 360° by 180° can be captured in about 40 s. Images are delivered in 96 bits per pixel TIFF RGB at a resolution of 100 Mpixels.

FIGURE 5.33 *An HDR panorama captured by Spheron's SpheroCam HDR line-scan camera, tone mapped using a histogram adjustment operator.*

5.12 CONCLUSION

In the not-too-distant future, digital still and motion picture photography may become exclusively HDR. After all, traditional film photography has provided medium dynamic range capture for nearly a century, and professionals expect and require this latitude during postproduction (i.e., printing). The current trend toward mixed reality in special effects is also driving the movie industry, which is increasingly digital, toward HDR. Advances in dynamic range will hit the professional

markets first and slowly trickle into the semiprofessional price range over a period of years.

Unfortunately, consumers will continue to be limited to LDR digital cameras in the short term, as HDR equipment will be priced out of reach for some years to come. During this interim period, software algorithms such as those described in this chapter will be the most affordable way to obtain and experiment with HDR imagery, and it is hoped that applications will push the market forward.

Display Devices and Printing Technologies

06

Some high dynamic range (HDR) images are captured and stored with the sole intent of using them as input for further processing such as image-based lighting (Chapter 11). However, in most cases, the ultimate goal is to present the HDR image to a human observer, either through a hard-copy medium such as a reflective print or slides (Section 6.3), or on a display device such as a television, computer monitor, or digital projector.

Because such display devices generate the photons that are ultimately processed by the human visual system, the characteristics of display devices have a major impact on the perceived quality of the reproduced image. In the following, we first give an overview of the most common display technologies before discussing in more detail the emerging dual-modulation (or local-dimming) technology that promises to deliver HDR images on displays and digital projection systems alike.

6.1 DISPLAY TECHNOLOGIES

Most conventional display technologies are low dynamic range (LDR) in nature, although some of the oldest devices in existence possess the potential for greater dynamic range than more recent technologies. We start with a discussion of one such device, which for all its problems has outlasted virtually every other electronic invention before or since: cathode ray tubes (CRTs).

© 2010, Elsevier Inc. All rights reserved.
DOI: 10.1016/B978-0-12-374914-7.00006-3

6.1.1 CATHODE RAY TUBES

The first electronic display was the CRT, invented by German physicist Karl Ferdinand Braun in 1897. A CRT is a vacuum tube configured to dynamically control the aim, intensity, and focus of an electron beam, which strikes a phosphor-coated surface that converts the energy into photons. By depositing red, green, and blue phosphors in a tight matrix and scanning the display surface at 25 Hz (Phase Alternate Line [PAL], Séquentiel couleur à mémoire [SECAM]), 30 Hz (National Television System Committee [NTSC]), or more, the eye can be fooled into believing it sees a continuous, two-dimensional color image. Each pixel is scanned once per frame, and the phosphor's gradual decay (coupled with the brain's integration of flashed illumination) makes the pixel appear as though it were constant. A good part of the success of the CRT is due to its inherent simplicity, although a century of tinkering has brought many variations and tens of thousands of patents to the basic technology.

In early computer graphics, CRTs were used in vector graphics, or *calligraphic* mode, in which continuous primitives such as lines or circles were drawn directly onto the screen by generating the appropriate control voltages for deflecting the electron beams. Today, almost all CRTs are operated in *raster* mode, where a single electron beam sweeps left to right, top to bottom, across a glass screen uniformly coated with a phosphor (see Figure 6.1), giving rise to the term "scanline."

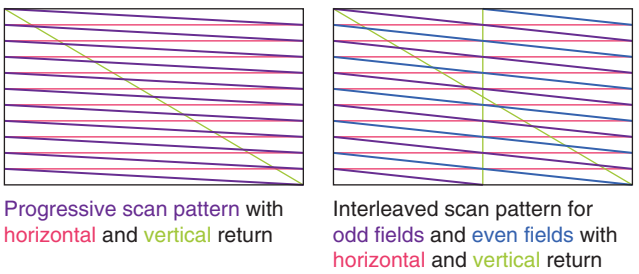

Progressive scan pattern with horizontal and vertical return
Interleaved scan pattern for odd fields and even fields with horizontal and vertical return

FIGURE 6.1 *Progressive (left) and interleaved (right) scanning patterns used in CRTs.*

For historical reasons, many video standards use a so-called "interleaved scanning pattern," in which the odd and even scanlines are traversed separately, in odd and even half-frames, called "fields." As the beam scans across the surface, its intensity is modulated continuously to produce different intensities. This design results in a curiosity of CRTs from the point of view of digital imaging: While a CRT represents an image as a discrete collection of scanlines, the horizontal intensity distribution is, in fact, a continuous function. As a result, the horizontal resolution is defined by the maximum frequency encoded in the video signal, as opposed to a hard limit defined by the CRT design.

For displaying color, the CRT design is modified to consist of *triads* of three different phosphors (red, green, and blue) where each color is addressed by its own electron gun. Figure 6.2 shows the anatomy of a CRT color triad in a conventional dot matrix layout. Other possible layouts include vertical stripes (e.g., Sony Trinitron). Although the interleaving of different-colored phosphors is in effect a discretization of the image plane, the color triads in a CRT should not be confused with pixels in the conventional sense. It is possible both for a pixel to be represented as multiple triads and for the electron beam to change intensity within the area of a triad. The mismatch between triads and pixels is one reason why CRTs do not produce images as sharp as those produced by other display technologies.

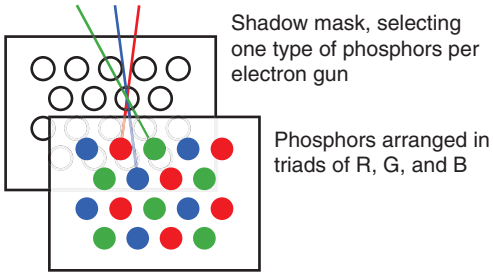


FIGURE 6.2 *A schematic of the dot masks used to selectively address different phosphors within a triad.*

Because of the use of phosphors, which may have fairly narrow emission spectra, CRT displays have some of largest color gamuts of all display devices. However, the phosphors are susceptible to "burn-in," or permanent degradation of the phosphors in areas where the same (bright) image content is displayed for extended periods of time. This phosphor aging also results in a degradation of CRT brightness and contrast over time.

Regarding dynamic range, the fundamental constraint for CRTs is their maximum brightness, which is limited by the amount of energy we can safely deposit on a phosphorescent pixel without damaging it or generating unsafe quantities of X-ray radiation. However, the maximum contrast of a CRT can be very high, as it is possible to switch off the electron beam completely in a local area. Unfortunately, most of this dynamic range lies at very low intensities, which are not discernible by a human observer under normal lighting conditions. As a result, CRTs are mostly used in very dark environments, both to limit reflections of stray light off the display surface and to allow the observer to adapt to a lower ambient light setting and thus increase sensitivity to low luminance values.

One approach to boost the dynamic range of CRTs is to combine raster scanning with a calligraphic mode [212]. Typically, such displays use raster scanning to produce regular image content, whereas calligraphic mode is used to let the electron beam dwell longer on a small number of high-brightness areas such as small light sources. The calligraphic mode could either use a separate electron gun or use the existing guns between frames. Mixed-mode raster and calligraphic CRTs were found primarily in flight simulators, and were supported by high-end graphics hardware such as the Silicon Graphics Reality Engine and programming interfaces such as Performer [78].

6.1.2 PLASMA DISPLAYS

Plasma panels are a flat display technology that, like CRTs, use phosphors for light generation. Unlike CRTs, however, plasma displays do not use space-consuming electron beams to excite these phosphors. Instead, the phosphors are coated inside little cells, three per pixel, which contain a noble gas such as argon, neon, or xenon. By applying a high voltage between the front and the back of each cell, the gas can be ionized, turning it into a plasma. As the ions of the plasma are attracted by the

electrodes, the ions collide and emit photons, primarily in the ultraviolet range. The phosphors within each cell convert this radiation into visible light.

This design allows for the construction of much thinner displays than would be possible with CRTs, although not as thin as liquid crystal displays (LCDs). As a result of the use of phosphors, the optical properties of plasma displays are similar to those of CRTs. They offer a wide color gamut, relative view independence, and relatively high-contrast ratios. The black level of plasma display is not quite as dark as that possible with CRTs because the cells need to be "precharged" to offer fast enough transitions from the dark to the bright state. However, plasma displays can be made brighter than CRT displays, as radiation is less of a concern.

On the downside, phosphor burn-in is also an issue with plasma displays, and their glass surface, much like that of CRTs, reflects more ambient light than other display technologies. Finally, plasma displays consume significantly more power than competing technologies. These disadvantages have in recent years led to a decline in plasma display sales and a shift toward LCDs.

6.1.3 LIQUID CRYSTAL DISPLAYS

LCDs represent a collection of flat-panel display technologies based on light filtering rather than emissive phosphors.

Liquid crystals (LCs) are elongated molecules that, depending on their orientation, can polarize the light passing through them. In the absence of external forces, LC molecules tend to align themselves with each other. In the presence of an electrical field, however, the LC molecules tend to align with the electric field lines. This property of LCs can be used to electronically manipulate the polarization direction of transmitted light. In combination with static polarizing sheets, LCs can be used to generate cells whose transparency can be controlled electronically. Arrays of such cells (pixels) can be placed over a uniform *backlight* module to form a display. The backlight is usually a combination of cold cathode fluorescent lamps (CCFLs) and light guides, which uniformly spread the illumination over the area of the display.

There are a number of different approaches for the specific way in which LCs are used to modulate light transmission. Figures 6.3 and 6.4 shows an overview of some of the most pertinent technologies, which present different trade-offs of properties such as cost, image quality, and switching speed.

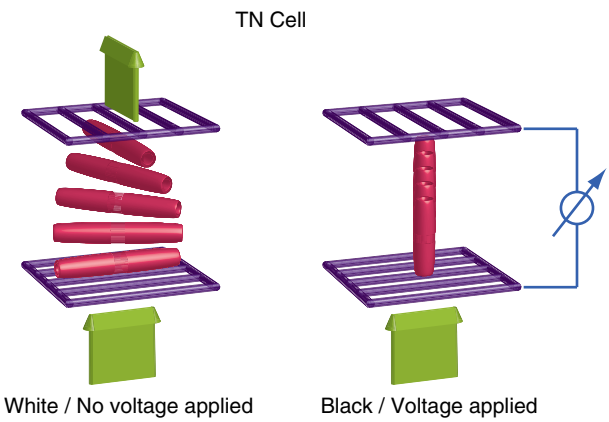


FIGURE 6.3 *Twisted nematic (TN) cell in the "black" state (left) and the "white" state (right).*

The simplest LCD design is the twisted nematic (TN) cell (Figure 6.3). In this design, the LCs are sandwiched between two polarizers and glass plates that have grooves on the inside surface. The grooves on the front surface are rotated 90° with respect to the grooves on the back surface. When no voltage is applied between the front and the back surface, the LC molecules align both with each other and the grooves on either glass surface, forming a twisted structure in three dimensions. In this way, light passing through the back polarizer is aligned with the LC molecules at the back surface. This polarization direction is rotated with the molecules so that the light exits at the front surface polarized in the orthogonal direction. The front polarizer is oriented in the same direction and lets this rotated light pass through. This is the "transparent," "white," or "on" state of the pixel.

The "black" or "off" state is achieved by applying a voltage between the front and the back surface with the help of a *thin film transistor*. In this situation, the LC molecules tend to align with the electric field lines rather than the grooves on the glass plate.

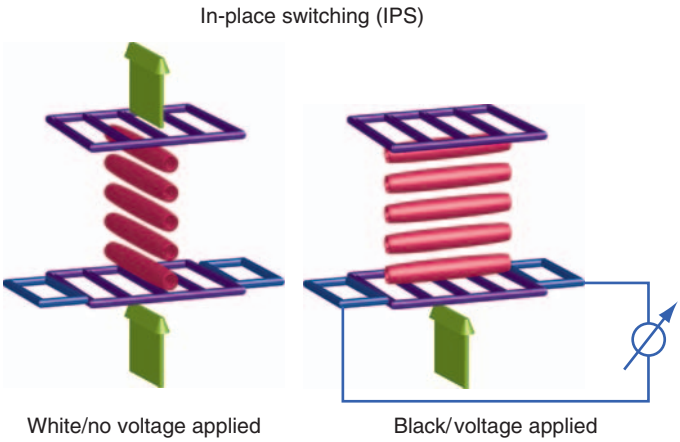


FIGURE 6.4 *In-plane switching (IPS) in the "black" state (left) and the "white" state (right).*

In this orientation, the molecules do not change the polarization direction of the light, which therefore gets absorbed by the front polarizer.

The TN cell design is both inexpensive and provides fast response times as low as 2 ms, making it the most commonly used LCD technology. A downside, however, is that the different orientations of the LC molecules at the front surface result in variations of the image with viewing angle. Although this situation can be mitigated by additional films on the front surface, it has also prompted the development of different technologies.

One of these is the multidomain vertical alignment (MVA) design Figure 6.3, in which each cell is divided into multiple zones of different surface orientations. Unlike TN cells, the default (no-voltage) state of an MVA display is "black," with the LC molecules aligned perpendicularly to the differently sloped zones. When a voltage is applied, the molecules rotate to change the polarization state and let the light pass.

The MVA technology is credited with much improved angular uniformity, while maintaining the high speed of the TN cell at moderately higher production cost. However, MVA compromises color reproduction and brightness.

In-plane switching (IPS) is a relatively new technology in which the LC molecules are kept parallel to both the front and back surface (Figure 6.4). Although this design results in superior angular uniformity, it suffers from slower response times and higher cost. Initial issues with color reproduction have since been resolved by refined designs.

Compared with CRTs and plasma displays, LCD technologies are capable of a higher brightness, simply by increasing the intensity of the backlight illumination. At the same time, the contrast of LCD panels is much more limited than that of the other two technologies, as it is difficult to achieve a very absorptive dark state. Both the polarizers and the LC materials are imperfect, resulting in light that is only partially polarized, and thus passes the panel even in the supposed "off" state. For these reasons, the contrast ratio of most LCD panels is around 300:1 for most displays, although some recent panels claim contrasts up to about 2000:1. Bear in mind that there are no well-respected industry standards for such measurements, and some manufacturers interpret contrast as the "all-on" versus "all-off" luminance ratio, while others use some type of checkerboard pattern, yielding more representative values.

Color in LCD panels is achieved by implementing three cells: LC cells with red, green, and blue color filters for each pixel. The color gamut of the final display depends on both the spectral characteristics of the backlight and the transmission characteristics of the color filters. The commonly used CFL backlights have a relatively flat spectrum over the visible range. This mandates the use of narrow color filters to achieve saturated primaries, and thus a wide color gamut. However, narrowing color filter spectra also reduces the overall brightness of the display, thus increasing the power requirements for the backlight for the same display intensity. The combination of opaque screen regions (regions covered by control circuitry and wires) as well as absorption in the color filters and the polarizers eliminate up to 95% of the light-emitted by the backlight.

A possible solution to this problem is the use of colored light-emitting diodes (LEDs) rather than CFLs in the backlight. Red, green, and blue LEDs can be combined into a white backlight unit that has three distinct spectral peaks rather than the

continuous spectrum of CFLs. As long as the color filters in the LCD panel are spectrally aligned with these peaks, a wide color gamut can be achieved with reduced absorption.

6.1.4 REFLECTIVE DISPLAY TECHNOLOGIES

The display technologies discussed so far produce their own light. Although this has some advantages for static displays, the energy consumption can be prohibitive for mobile devices, especially when the device needs to be bright enough to be legible in direct sunlight. An alternative to such devices are reflective display technologies, which use ambient light to produce an image.

Electronic Paper. Electronic paper refers to a collection of reflective technologies, including but not limited to *electrophoretic* displays. Electrophoretic displays contain small white particles made of materials such as titanium dioxide as well as absorbing dyes. Either the white particles or the absorbing dyes (or both) are charged, so that they can be moved toward or away from the front surface of display by the application of an electric field (electrophoresis). The mixture of reflective and absorptive particles at the display surface determines the reflectivity of the display at that location. Electrophoretic displays represent the most commonly used electronic paper technology, and are used, for example, in cell phones and e-book readers. However, there is a host of alternative electronic paper technologies in various stages of development. They typically combine a number of characteristics, including being bistable (only using power when changing the display content) and black-and-white or monochromatic. Current electronic paper displays are also slow, often requiring over a second for an image update.

Reflective LCD Panels. Reflective LCD panels use LC technology to modulate reflected rather than transmitted light. Many mobile devices such as navigation devices not only use LCD panels that can be operated in reflective mode, but also have a backlight for dark environments. These are called "transflective" displays. Although reflective and transflective LCD displays typically have color, the reliance on ambient light makes color reproduction unpredictable. Such displays are thus most often found in mobile devices used for information display (cell phones or navigation devices), as opposed to devices where image quality matters (cameras or mobile video players).

From the point of view of dynamic range, reflective display technologies remain significantly behind the transmissive and emissive technologies mentioned earlier. This is primarily because it is hard to obtain very deep absorption (black) with reflective technologies. Some electronic paper technologies produce contrast ratios as low as 10:1. Reflective LCD panels have better contrast than electronic paper, but stay far behind transmissive LCD panels, as well as CRTs and plasma displays.

6.1.5 PROJECTION TECHNOLOGIES

Projection systems are displays that focus an image onto a projection screen by means of an optical system. This class of display includes a large range of formats, from rear-projection televisions, in which the projector and screen are integrated in a single housing, to the projectors used in meeting rooms, and high-end systems used in digital cinema projection. The basic principle, however, is the same for all these devices: Light from a bright source such as a xenon lamp is collimated, and then filtered (modulated) through either a reflective or a transmissive display such as an LCD panel. The filtered light is reimaged onto the projection screen using an objective lens.

In projection systems, the individual displays used to modulate the light from the bulb are almost always monochromatic. There are two primary means for achieving color projection. The first, most commonly used in consumer devices, is *field sequential color*, in which the red, green, and blue channel of the image are projected one after the other by passing the light through different color filters. Most commonly, these filters are mounted onto a rotating wheel (the *color wheel*) that spins in front of the light source. Field sequential color is known to cause a disturbing "rainbow" artifact, where supposedly mixed colors break up into spatially distinct regions during rapid head or eye movement. These artifacts can be partially mitigated by increasing the rate at which the color fields are projected.

The second, more costly, design is to use three separate displays, one each for the red, green, and blue channel. In this design, the light from the bulb is first split up into different color components using *dichroic filters*, and the red, green, and blue light are then modulated by individual displays in individual light paths. Finally, the three components are recombined and projected by a common objective lens.

Although early video projectors used LCD panels very much like the ones in desktop displays and television screen, most modern projectors use specially developed display technologies that offer a higher resolution at a small form factor.

Liquid-Crystal-on-Silicon. Liquid-crystal-on-silicon (LCoS) refers to a reflective LCD technology where the LC is directly coated onto a silicon chip. The limited display size imposed by this technology is not an issue and, in fact, is desired for projection displays. Compared with other LCD technologies, LCoS allows for smaller pixel sizes and, therefore, higher-resolution displays at the same physical dimensions. Most LCoS projectors use a three-chip design for color projection.

Micro-Mirror. Micro-mirror projectors, also known as "digital light-processing" (DLP) projectors, rely on a microelectromechanical system consisting of small mirrors that can be electronically tilted into one of two positions at very fast rates. Every mirror corresponds to one pixel, and one mirror position redirects light from the light source to the lens and on to the projection screen, while the other mirror position directs the light toward an absorbing surface. Gray intensity values (and color gradations) are obtained by rapidly altering the orientation of the mirror within a single frame. Most consumer DLP projectors use a field sequential color approach, whereas digital cinema projectors use a three-chip design.

6.1.6 EMERGING DISPLAY TECHNOLOGIES

In addition to the display technologies described so far, there are a host of prototype and experimental technologies at various stages of development. Describing each of these in detail would exceed the scope of this book. However, we would like to briefly discuss two more technologies, which have gained some momentum and are seen by some as promising future replacements of current technologies.

Organic Light-Emitting Diode. Organic light-emitting diode (OLED) displays are panels in which each pixel consists of a separate, small LED made from organic compounds. In principle, this design allows for a large color gamut and very HDR by providing both high brightness and a dark "off" state at relatively low power consumption. OLED displays are already in widespread use in small form factors in cell phones, cameras, and other mobile devices. The biggest challenges for OLED displays at this

time remain the high cost of the technology, difficulties in producing large form factors and a very limited lifetime.

Although these problems can probably be overcome in time, doing so requires major investments in the technology. Whether such investments are justified in light of continuous improvements of LCD panel technologies and the novel dual-modulation concepts we discuss in the next section is still being debated in the display industry.

Laser Displays. Laser display is a technology that has been worked on for several decades. As in the case of OLEDs, the key promise of laser displays is a wide color gamut combined with very HDR. The key technical challenge are disturbing interference patterns, called ''laser speckle,'' which are created by the coherent nature of laser light. Because the interference patterns are created inside the human eye, they cannot be prevented by technical means without sacrificing the coherence of the laser light.

6.2 LOCAL-DIMMING HDR DISPLAYS

One of the most promising display technologies for true HDR displays is *local-dimming displays*, often also called "dual-modulation displays." The basic principle behind this approach is to optically combine two spatial light modulators (displays) in such a way that their contrast multiplies [287]. A simple example of this concept is the combination of a video projector and a transmissive LCD panel, as shown in the image of a prototype in Figure 6.5. The schematic of such a display is shown in Figure 6.6.

The video projector in this example is based on DLP technology, but any other projection technology would work as well. The image of the projector is collimated with a Fresnel lens, and projected onto the back surface of a transparent LCD panel. Thus, the projector image acts as a "backlight" for the LCD panel. It is easy to see that this optical setup approximately multiplies the contrast of the video projector with the contrast of the LCD panel: Let the projector contrast be $I_w : I_b$, where I_w and I_b are the intensity of the projector "white" and "black," respectively. Likewise, let the LCD panel contrast be $\alpha_w : \alpha_b$, where α_w and α_b are the transparencies of the LCD panel in the "white" and "black" state, respectively. It is easy to see that the

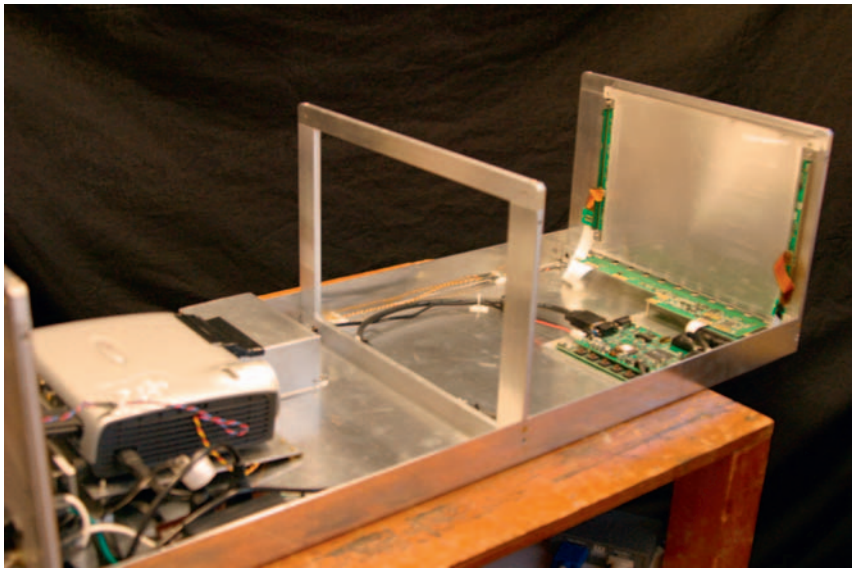


FIGURE 6.5 *One of the first dual-modulation prototype displays. Reproduced from Seetzen et al 2004 (Siggraph)*

combined device can produce a darkest intensity ("black") corresponding to $\alpha_b \cdot I_b$, and a brightest intensity corresponding to $\alpha_w \cdot I_w$.

In practice, the theoretical contrast is not quite achieved because of optical imperfections such as scatter in the various components, but contrast ratios in excess of 100 000:1 have been demonstrated with this setup, along with brightness values up to 5000 cd/m^2, or about 10 times the brightness of a regular display.

Although the projector/LCD panel combo demonstrates the basic merits of the dual modulation concept, it has some obvious limitations. First, the optical path of the projector requires a very deep display form factor. Also, power consumption, heat, and maintaining good alignment between the projector and LCD panel are

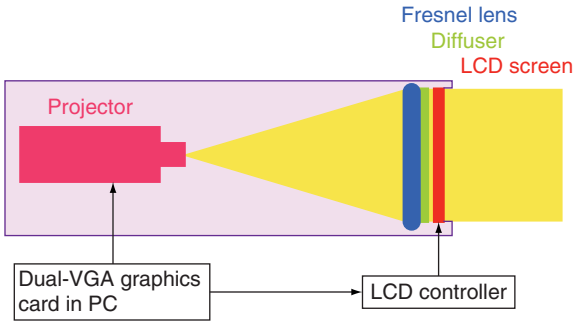


FIGURE 6.6 *Diagram of the projector/LCD panel realization of the dual-modulation or local-dimming concept.*

significant practical obstacles. Nonetheless, this particular setup is useful for psychophysical studies [180] and to simulate different configurations for the second modulator [285]. The display in Figure 6.7 has been built with the latter application in mind.

More practical realizations of the dual-modulation concept are possible by taking into account results from psychophysics, and in particular the effect of *veiling glare* (also see Sections 8.4.2 and 10.7.1). "Veiling glare" refers to the *local* reduction of contrast on the retina because of scattering in the eye (lens and ocular fluid). This effect, which has been well documented and measured, primarily impacts the visibility of dark detail near bright features.

Veiling glare means that the contrast that can be processed by the human visual system *locally* around an edge is much more limited than the *global* contrast that can be processed over a wider range of directions. From a display point of view, this means that it is not necessary to be able to produce very dark pixels directly adjacent to very bright ones, as long as both intensities can be accurately portrayed in *different* image regions.

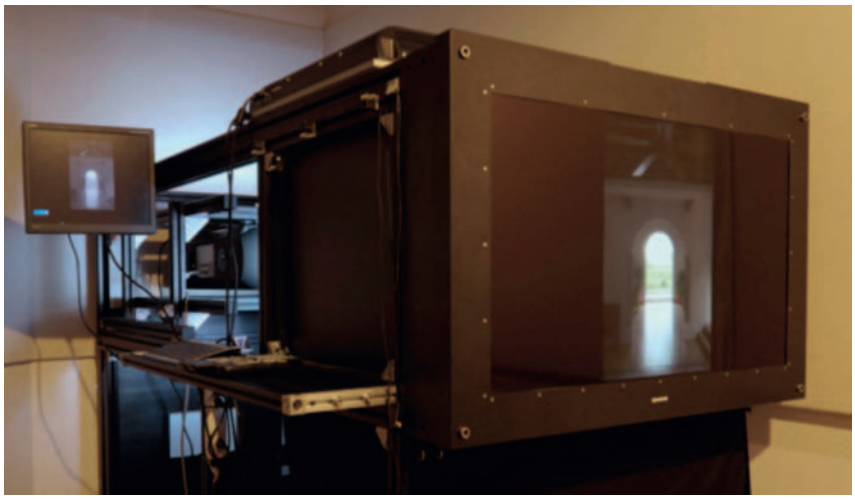


FIGURE 6.7 *A more recent projector/LCD panel setup used for the experimentation with system parameters.*

6.2.1 LOCAL-DIMMING TELEVISION SCREENS

This key insight gave rise to dual-modulation HDR displays where one of the modulators has a significantly lower resolution than the other. In the latest generations of commercially available LCD television sets, this concept has been implemented by replacing the uniform backlight of the LCD panel with a low-resolution grid of LEDs (typically around 2000 for a 2-Mpixel display), which can be controlled individually or in clusters. Instead of producing light uniformly over the whole image region, the LEDs are dimmed or off behind image regions of reduced brightness. This has led to the term "local-dimming" for these types of displays.

The advantages of this construction over the projector/LCD approach are obvious: the local-dimming approach allows for thin display form factors, and has

significantly reduced energy requirements (and thus heat production) compared with the earlier approach.

At the time of this writing, most commercially available local-dimming displays are not significantly brighter than normal televisions, although they do offer much improved contrast ratios. The DR 37P, a small series of true HDR display based on this technology, was built by the startup company Brightside Technologies, which was acquired by Dolby Labs in 2007. The DR 37P, which has been used for a large number of perceptual studies and is still in use by research labs around the world, offered peak intensities of 4000 cd/m^2 with a simultaneous checkerboard contrast of over 150 000:1. Recently, Italian manufacturer Sim2, in collaboration with Dolby, has announced a commercial display with similar performance characteristics.

6.2.2 DUAL MODULATION IN LARGE-FORMAT PROJECTION

Although dual-modulation technology is closer to commercial deployment in the television market, there are also opportunities to apply this technology in large-screen projection. A number of these possibilities have been outlined by Seetzen in his Ph.D. thesis [286]:

Projector-Integrated Second Modulator. The most straightforward implementation of the dual-modulation concept in projectors is to integrate a second modulator, such as an LCD panel directly into the projection unit. As before, this second modulator can be monochromatic, and have a much reduced resolution. A prototype of this approach was demonstrated by Damberg et al. [52]. This prototype demonstrated convincing improvements in terms of dynamic range, but at the cost of lower light throughput because of the extra absorption in the second modulator. The integrated second modulator in a projector lacks one of the key advantages of the local dimming LED backlights: Light is produced uniformly everywhere over the image plane and then absorbed, rather than only being produced where it is required. This poses additional challenges in terms of power consumption and heat management in projector design.

Screen Modulation. A second approach proposed by Seetzen, and independently by Bimber and Iwai [21], is to turn the projection screen into a low-resolution second modulator, for example, by making it out of a material with programmable reflectance, such as electronic paper. This setup is somewhat similar to the initial projector/LCD panel HDR prototype display, except that the projector is now projecting a high-resolution color image, and the projection surface is low-resolution and monochromatic.

A key advantage of the screen modulation approach is that electronic paper-like materials can achieve a relatively high reflectivity, so that the projector intensity can remain unaltered compared with current designs. Another advantage of dimming the screen itself is that this also suppresses the impact of ambient light, which would otherwise wash out dark regions of the image. The key obstacle to the deployment of this method at the moment is the slow update rate of electronic paper technology and its high cost. The prototypes of this technology are currently not video-capable, and only show static images.

Tiled Projector Concept. Finally, Seetzen proposes the use of a large number (hundreds or even thousands) of small, very inexpensive projectors. Each projector would have an inexpensive, low-resolution LCD panel; a single, dimmable LED for a light source; and simple plastic focusing optics. Seetzen estimates that such a projector could be built in volume for less than US$10. The projectors would be set up such that their projections partially overlap on the screen. The geometric alignment of all components would be calibrated with a camera.

In this configuration, the dimming of the individual projector LEDs provides the second modulation. This approach provides the same benefits as the local-dimming LED backlights: Light is only produced in image regions where it is required. Seetzen demonstrated the basic principle of this method on a configuration with a small number of projectors, but a full system is yet to be built.

6.2.3 IMAGE PROCESSING FOR DUAL-MODULATION SCREENS

Independent of which variant of the dual-modulation concept is used, displaying an HDR image on a dual-modulation screen requires significant image-processing

efforts to select the "pixel" values for the two modulators. In the following, we will use the terminology of the local-dimming TV technology for simplicity. However, the other dual-modulation approaches require similar algorithms.

Given an HDR input image, the display driver for a local-dimming display must choose LED intensities and corresponding LCD pixel values (transparencies) to best represent the image. Because these calculations depend on the LED layout and optical characteristics (point spread functions, transmission curves, etc.) of the display, this separation into LED and LCD components cannot be done as a preprocess earlier in the video pipeline, but has to be done by the display itself, and in real-time. In practice, compromises between image quality and computational efficiency are therefore required.

The task at hand can fundamentally be described as an optimization problem, where the LED and LCD values are to be determined such that the image produced on the HDR display has the smallest possible *perceptible* difference from the original HDR image [324,323]. The optimization is also subject to *physical constraints* such as maximum and minimum intensity/transparency.

Because computational models of human perception are too expensive for real-time computations, a more practical approach is to develop heuristic algorithms and to validate their performance using psychophysical studies on a large range of test images and videos. One simple heuristic algorithm is depicted in Figure 6.8 [287].

The algorithm starts by splitting the original HDR image into two "target" components, corresponding to an allocation of contrast between the LEDs and the LCD panel. The square-root function splits the target contrast equally between the LEDs and the LCD panel. Other power functions or different splits are also possible.

Once a target image for the LED backlight has been determined in this fashion, the best approximation of that image with the low-resolution LED grid has to be computed. This is done by downsampling the target image to the LED resolution, and approximately deconvolving by the LED point spread function. The results of this operation are the LED values, which are directly sent to the display hardware.

To compute the best "compensation" pixel values, for the LCD panel, we first have to simulate the *actual* light distribution generated by the LEDs. This is done by convolving the chosen LED value with the LED point spread functions at the full image resolution. Dividing the original HDR image by this LED pattern gives the LCD transparency values required to reproduce that image. These transparencies need to

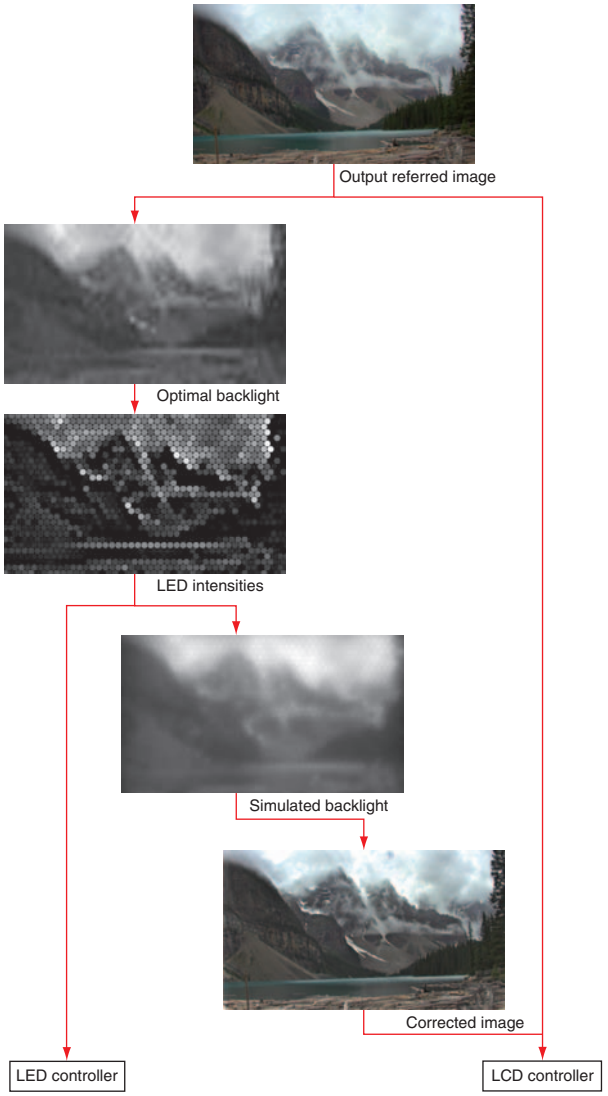


FIGURE 6.8 Block diagram of a heuristic image-processing algorithm for HDR displays.

be converted to pixel values, by applying the LCD response function, before actually driving the display.

A number of variations of this basic approach have been proposed over time (e.g., [37,117]). Recent modifications include algorithms for driving displays with colored LEDs, as well as finding trade-offs between image quality and power consumption [324].

6.3 PRINTING

The first image-duplication systems were hard-copy devices, going all the way back to Johann Gutenberg's invention of movable type and oil-based inks for the printing press in the fifteenth century. This was truly a digital device, requiring dexterous fingers to place the letters and designs on the master plates. (Wood-block printing dating back to eighth-century China was more of an engraving transfer process.) Hand presses eventually gave way to the powered flatbed cylinder presses in the 1800s, which are still used for many printing applications today. More significant to this discussion, the dawn of photography in the latter half of the same century opened a new horizon not only to the printing process but also to what could, in fact, be printed.

Significantly, the chemistry of black and white film, and later color-negative stock, have been tailored to record HDR information. As discussed in Chapter 5, the photographic printing/enlargement process is where the original range of the negative is reduced to fit the constrained range of a standard reflection print. The additional depth in the shadow and highlight areas of the negative permit the photographer or the processing lab to perform adjustments to the image exposure a posteriori to optimize the final image. This was the original use of the term "tone mapping," now recognized to be so important to computer graphics rendering [326].

Figure 6.9 shows a color negative of an HDR scene next to a typical LDR print. The false color image on the right shows that the range recorded by the negative is actually quite large—nearly four orders of magnitude, and some information in the shadows is lost during standard printing. Using dodge-and-burn techniques, a skilled darkroom specialist could bring these areas out in a handmade print.

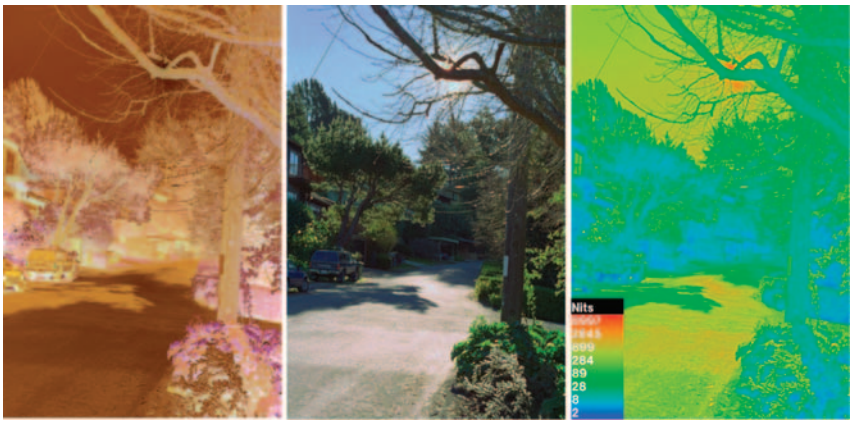


FIGURE 6.9 *A color photograph of an HDR scene. The negative shown on the left stores scene luminance with its native logarithmic response. The middle image shows an LDR print, whereas the right image shows the actual range available from the negative.*

By scanning the full dynamic range of the negative, one could alternatively apply one of the latest digital tone-mapping operators to compress this information in a lower dynamic range output. This fits with the idea of storing a "scene-referred" image and applying a device-dependent tone mapping prior to final output. (See Chapters 7 and 8 for further information.)

6.3.1 THE REFLECTION PRINT

As implicitly illustrated in all the figures of this and every other book, reflective print media is inherently LDR. Two factors are responsible for this. First, the brightest pixel in a reflection print is dictated by the ambient lighting. This same ambient

light illuminates the area around the print, which we can generally assume to be a medium color (midgray being 18% reflectance, but see the footnote on p. 64). Thus, even the whitest paper stock with a 90% reflectance is only perhaps five times as bright as its surroundings. A typical specular highlight on a sunny day is 500 times as bright as its surroundings, and light sources can be even brighter. Would it be possible to represent these outstanding highlights in a reflection print? One possible solution is to selectively use a glossier material to represent high-brightness regions. Under the right lighting conditions, glossy materials direct most of the reflected light toward the viewer, rather than spreading it uniformly over all directions, like (mostly diffuse) paper. Early artists already recognized this solution and added gilding to their paintings and manuscripts [102], but this would be unreliable (not to mention expensive) in a commercial print setting.

The second limitation to the contrast of reflection prints is the maximum absorption, which is generally no better than 99.5% for most dyes and pigments. Even if we had a perfectly absorbing ink, the surface of the print itself reflects enough light to undermine contrast in the deep shadow regions. Unless the illumination and background are very carefully controlled, the best contrast one can hope for in a good viewing environment is about 100:1, and it is often much less.

Figure 6.10 shows a chart of different print *density*, where density is defined as $\log_{10}(1/R)$, and R is the reflectance. Adjacent bars in the chart differ by roughly 11%—well above the visible difference threshold, and spaced for optimum visibility. Even though we have given the image a black background to improve contrast visibility, the steps become indistinguishable well before a density of -2 (1% reflectance). On an HDR display, these steps would be clearly visible all the way to the bottom of the chart. The fact that they are not demonstrates one of the inherent limitations of diffusely reflective media—LDR output.

An emerging solution to HDR "reflective" print is the use of fluorescent inks. The fluorescent dye in those inks converts ultraviolet illumination into different visible wavelengths, giving the impression of a "brighter-than-white" surface, with the possibility for very vibrant colors. Some artists already use fluorescent dyes in hand-painted images to represent scenes such as sunsets (e.g., [311]). Recently, there has also been some work on modeling the color space of fluorescent inks for printing applications [126,125], although to our knowledge, there has not yet been a lot of work on combining fluorescent and reflective inks.

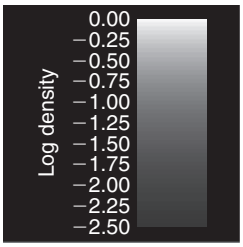


FIGURE 6.10 *A density chart, demonstrating that it is difficult to resolve differences at or below 1% reflectance (-2.0 $\log_{10}$ density) in printed images.*

6.3.2 TRANSPARENT MEDIA

Not all hard-copy media are reflective—some media are transparent and designed to be projected. The most obvious example is movie film, although 35-mm slide transparencies and overhead transparencies Bear mentioning as well. Fundamentally, transparencies overcome the two major limitations of reflective media—ambient lighting and maximum density. Because transparencies rely on a controlled light source and optics for display, the ambient environment is under much tighter control. Most transparencies are viewed in a darkened room, with a dark surrounding. For maximum density, we are only limited by film chemistry and printing method as to how dark our transparency can get—three orders of magnitude are regularly produced in practice, and there is no physical limit to the density that can be achieved.

Are slides and movies really HDR? Not really. They certainly have more dynamic range than standard reflection prints—perhaps by as much as a factor of 10. However, viewers prefer higher contrast for images with a dark surround [83], so the filmmakers oblige by creating high-contrast films for projection. The sensitive dynamic range of slide transparency film is actually quite narrow—about two orders of magnitude at most. Professional photographers are well aware of this

phenomenon—it is imperative to get the exposure and lighting exactly right, or there is no advantage in shooting transparency film. Cinematographers have a little more room to move because they go through an additional transfer step where the exposure can be adjusted, but the final print represents only a narrow range of luminances from the original scene.

Although transparency film is not traditionally used as an HDR medium, it has this potential. Something as simple as a slide viewer with a powerful backlight could serve as a low-tech HDR display if there were some way to produce a suitable transparency for it. An example of such an approach is demonstrated in the following section.

6.3.3 WARD'S HDR STILL IMAGE VIEWER

The dual-modulation concept introduced in this chapter can also be used in hardcopy media. Figure 6.11 shows an HDR still-image viewer composed of three elements: a bright, uniform backlight; a pair of layered transparencies; and a set of wide-field, stereo optics. The view mapping for the optics and the method

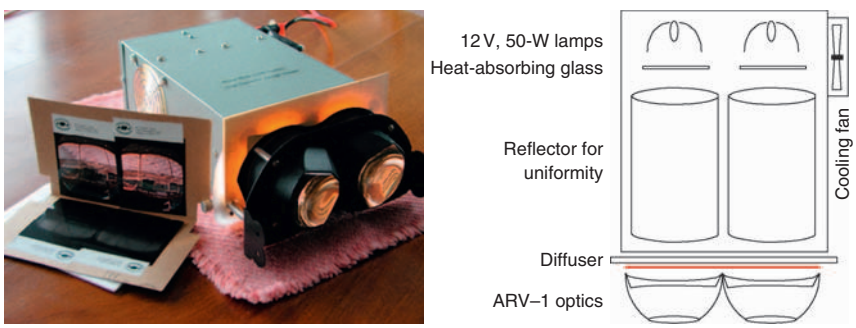


FIGURE 6.11 *An HDR viewer relying on layered film transparencies. The transparency position is shown in red on the right-hand diagram.*

for increasing dynamic range by layering transparencies are the two challenges faced [344]. The original prototype of this HDR viewer was built in 1995 to evaluate HDR tone-mapping operators, but it has only been recently been put to this task [179]. In the configuration shown, the viewer provides a nearly 120° field of view, a maximum luminance of 5000 cd/m^2, and a dynamic range of over 10 000:1. It uses the Large Expanse Extra Perspective (LEEP) ARV-1 optics, which were designed by Eric Howlett and used in the original NASA virtual-reality experiments [79].[1]

The LEEP ARV-1 optics use a *hemispherical fisheye projection*, where the distance from the center of the image is proportional to the sine of the eccentricity (i.e., the angle from the central view direction). Also, the optics exhibit significant chromatic aberration, which will cause colored fringes at the edges of view. This was originally corrected for by a matched camera with chromatic aberration in the opposite direction, but because we seek to render our views on a computer, we apply an equivalent correction during image preparation, the *Ca*() function described below. The image must be of high resolution not to appear blurred in the viewer—we found a resolution of 800 dpi (2048×2048) to be the minimum. A 4×5 film recorder is essential to produce transparencies at this size and resolution.

A film recorder typically consists of a small, slow-scan CRT with a white phosphor, which is carefully scanned three times with each of three colored filters interposed between the CRT and a film camera with a macro lens. The process is slow and the equipment is increasingly rare, making the production of high-resolution transparencies a costly proposition. As the LEEP optics require a 2.5"×5" transparency pair, we must split the job into two 4"×5" outputs because film cannot be printed to its borders. Furthermore, because of the difficulty of controlling transparency exposures to achieve densities where the film response is highly nonlinear, it is necessary to create two transparency layers per eye, doubling the cost again.[2]

Figure 6.12 shows the method for splitting a single HDR image into two transparency layers, which will later be mounted one atop the other in the viewer. This process is very similar to the one discussed in Section 6.2. Even though the two modulation layers use identical slide material, one of the modulation layers is deliberately

1 The ARV-1/diffuser assembly was obtained from Ulrecth Figge of Boston, MA.

2 This brings the total cost per view to around US$200.

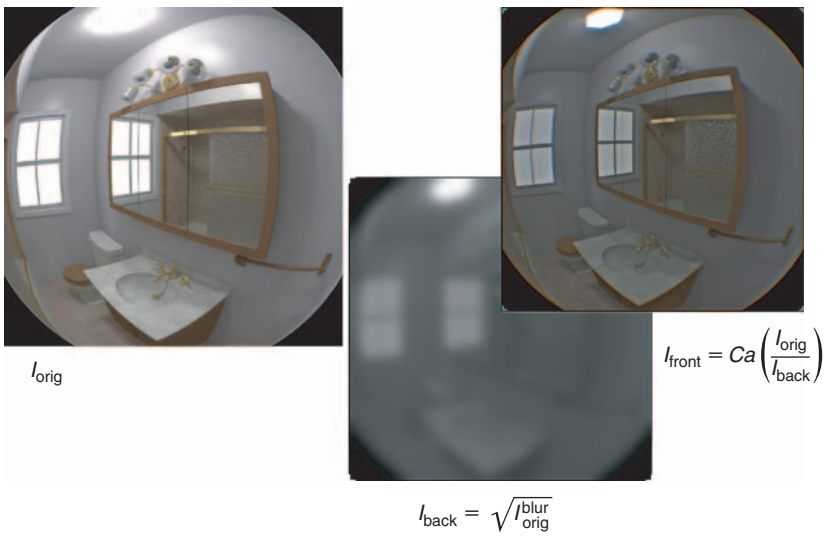


FIGURE 6.12 *The process whereby the original fisheye image for one eye is split into two transparency layers, which when combined in the HDR viewer will reproduce the original luminances.*

blurred, thus providing a modulation layer very much like the low-resolution LED grid in local dimming displays. This blurring is an essential trick for Ward's viewer, as it makes the alignment of the two slide layers feasible.

6.4 CONCLUSIONS

Although the reality of HDR reflection prints seems remote, the future of displays is more promising. Already there are dual-modulation displays on the market capable of high-contrast ratios, and with LEDs getting progressively cheaper and brighter,

LCDs with LED backlight arrays may bring HDR to the consumer television market before it reaches the local multiplex. In some ways, this is unfortunate, as HDR content will lag because of lack of impact at the box office, but inverse tone-mapping should hold us over until HDR movies begin to arrive (see Chapter 9). Eventually, we predict that HDR projection will have the biggest impact in the theaters because of surround sound, and one only has to see it to believe it.

This page intentionally left blank

Perception-Based Tone Reproduction

07

The dynamic range of illumination in a real-world scene is high, of the order of 10,000 to 1 from highlights to shadows, and higher if light sources are directly visible. A much larger range of illumination can also occur if the scene includes both an outdoor area illuminated by sunlight and an indoor area illuminated by interior light (see, e.g., Figure 7.1). Using techniques discussed in Chapter 5, we are able to capture this dynamic range with full precision. Unfortunately, most display devices and display media available to us come with only a moderate absolute output level and a useful dynamic range of less than 100 to 1. The discrepancy between the wide ranges of illumination that can be captured and the small ranges that can be reproduced by existing displays makes the accurate display of the images of the captured scene difficult. This is the high dynamic range (HDR) display problem or HDR tone-mapping problem. We introduce the tone-mapping problem in this chapter and discuss individual solutions in detail in the following chapter.

7.1 TONE-MAPPING PROBLEM

For a display to exhibit realism, the images should be faithful visual representations of the scenes they depict. This is not a new problem. Artists and photographers have been addressing this problem for a long time. The core problem is that the light intensity level in the environment may be beyond the output level reproduced by the display medium. Also, the contrast experienced in a real environment may

© 2010, Elsevier Inc. All rights reserved.
DOI: 10.1016/B978-0-12-374914-7.00007-5

FIGURE 7.1 *Image depicting both indoors and outdoors areas. The different lighting conditions in these areas give rise to an HDR. This image was tone mapped with the technique of Li et al. [186], which is discussed in Section 8.3.5.*

greatly exceed the contrast range that can be reproduced by those display devices. This affects a painter's canvas, a photographer's positive print, and a viewer's display device.

The appearance of a scene depends on the level of illumination and the contrast range [82]. Some commonly noticed examples are as follows: "Scenes appear more colorful and contrasty on a sunny day," "Colorful scenes of the day appear gray during night," and "Moonlight has a bluish appearance." Hence, simple scaling or compression of the intensity level and the contrast range to fit them into the display limits is not sufficient to reproduce the accurate visual appearance of the scene. Tumblin and Rushmeier [326] formally introduced this problem, depicted in Figure 7.2, and suggested the use of visual models for solving this problem.

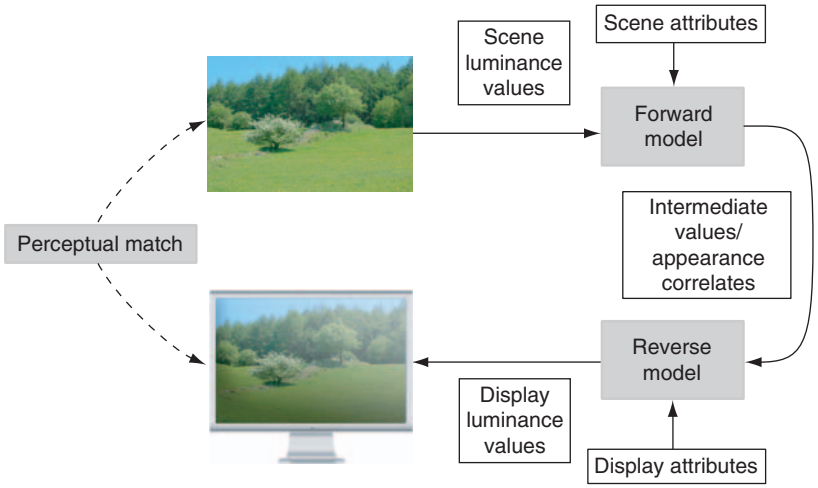


FIGURE 7.2 *Outline of the tone-mapping problem. The ultimate goal is a visual match between the observed scene and the tone-mapped version of the captured HDR image on the display. (Image courtesy of Tania Pouli)*

The use of visual models has become part and parcel of the development of tone-reproduction operators ever since.

Reproducing the visual appearance is the ultimate goal in tone mapping. However, defining and quantifying visual appearance itself is a challenge, and currently an open research problem. They have given rise to both color appearance models, which are discussed in Chapter 2, and image appearance models, which are presented in Section 8.2. In this chapter, we will address several basic issues for the realistic display of HDR images.

For the most part, we focus on the compression of luminance values to make the dynamic range of the image fit the range of a given display device. This can be achieved by simple scaling of the image. However, such simple scaling often

FIGURE 7.3 *HDR image depicting both indoors and outdoors areas. Linear scaling was applied to demonstrate the lack of detail afforded by linear scaling. Compare with the tone-mapped result shown in Figure 7.1.*

generates images with complete loss of detail (contrast) in the resulting display (Figure 7.3). As such, reducing the dynamic range of an image to fit the display is, by itself, not sufficient. We will also have to preserve some aspect of visual appearance. It is, for instance, possible to assume that the loss of detail ought to be avoided. This observation gives us a seemingly simpler problem to solve: How do we compress the dynamic range of the HDR image to fit into the display range while preserving the detail?

The human visual system (HVS) deals with a similar problem. The signal-to-noise ratio of individual channels in the visual pathway is about 32 to 1, less than two orders of magnitude [69,134]. Even with this limited dynamic range, the HVS functions well: It allows us to operate under a wide range of illumination, and gives

us the ability to simultaneously perceive the detailed contrast in both the light and dark parts of an HDR scene.

Thus, if the goal is to match this perceived realism in the display of HDR images, it is important to understand some of the basics of the HVS. Hence, this chapter focuses on aspects of the HVS relevant to HDR imaging. We show that most tone-mapping algorithms currently available make use of one of a small number of visual models to solve the HDR problem.

The material described in the following sections have been distilled from the psychophysics and electrophysiology literature, where light is variously measured as quanta, intensity, luminance, radiance, or retinal illuminance. To avoid confusion, wherever possible, we will use the term "luminance." If any unit other than luminance is intended, we will provide the units in which they appeared in the original literature.

7.2 HUMAN VISUAL ADAPTATION

A striking feature of the HVS is its capability to function over the huge range of illumination it encounters during the course of a day. Sunlight can be as much as a million times more intense than moonlight. The intensity of starlight can be one-thousandth of the intensity of moonlight. Thus, the effective range of illumination is more than a billion to one [334]. The dynamic range simultaneously available in a single scene at a given time is much smaller, but still hovers around four orders of magnitude.

The visual system functions in this range by adapting to the prevailing conditions of illumination. Thus, adaptation renders our visual system less sensitive in daylight and more sensitive at night. For example, car headlights that let drivers drive at night mostly go unnoticed in daylight, as shown in Figure 7.4.

In psychophysical studies, human visual adaptation is evaluated by measuring the minimum amount of incremental light that enables an observer to distinguish a test object from the background light. This minimum increment is called a "visual threshold" or "just noticeable difference" (JND).

In a typical threshold-measurement experiment, a human subject focuses on a wide blank screen for a sufficient amount of time to adjust to its uniform background intensity, I_b. Against this uniform background, a small test spot of intensity $I_b + \Delta I$

FIGURE 7.4 *Although the headlights are on in both images, our eyes are less sensitive to car headlights during daylight than at night.*

is flashed. This test spot is called the "stimulus." The increment ΔI is adjusted to find the smallest detectable ΔI_b. The value of this threshold depends on the value of the background, as shown in Figure 7.5, which plots typical threshold-versus-intensity (TVI) measurements at various background intensities.

Over much of the background intensity range, the ratio

$$\frac{\Delta I_b}{I_b}$$

is nearly constant, a relation known for more than 140 years as "Weber's law." The value of this constant fraction is around 1% [334] and can vary with the size of the test spot and the duration for which the stimulus is shown. The constant nature of this fraction suggests that visual adaptation acts as a normalizer, scaling scene intensities to preserve our ability to sense contrasts within it.

Visual adaptation to the varying conditions of illumination is thought to be possible through the coordinated action of the pupil, the rod–cone system, photochemical reactions, and photoreceptor mechanisms. The role of each of these factors is discussed in the following sections.

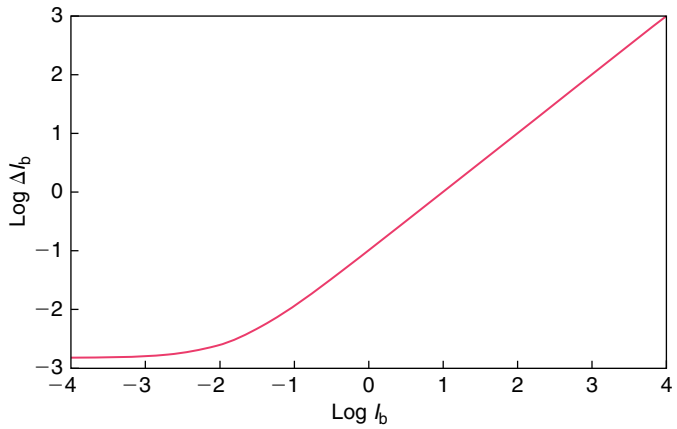


FIGURE 7.5 *TVI relation: The plot shows the visual threshold ΔI_b at different background intensities I_b.*

7.2.1 THE PUPIL

After passing through the cornea and the aqueous humor, light enters into the visual system through the pupil, a circular hole in the iris [97,122,128] (Figure 7.6). One of the mechanisms to adapt to a specific lighting condition is to regulate the amount of light entering the eye by adjusting the size of the pupil. In fact, the pupil changes its size in response to the background light level. Its diameter changes from a minimum of around 2 mm in bright light to a maximum of about 8 mm in dark. This change accounts for a reduction in light intensity entering the eye by only a factor of 16 (about one log unit). In a range of about 10 billion to 1, the intensity regulation by a factor of 16 is not very significant. Hence, the pupil's role toward visual adaptation is often ignored for the purpose of tone reproduction.

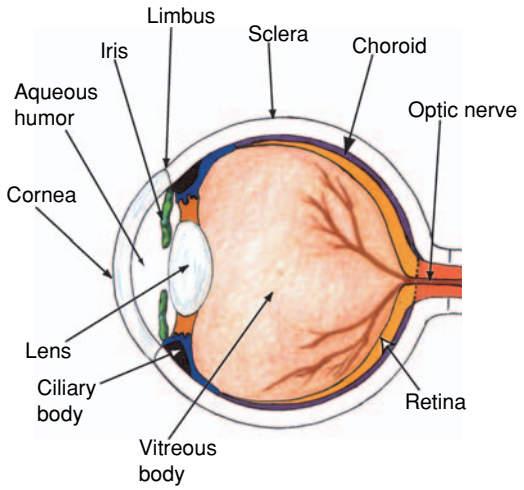


FIGURE 7.6 *Schematic diagram of the human eye and its various components. (Image courtesy of Karen Lefohn)*

7.2.2 THE ROD AND CONE SYSTEMS

Light that passes through the pupil travels through the lens and the vitreous body before reaching the retina, where it is reflected off a pigmented layer of cells before being absorbed by the photoreceptors. The latter convert light into neural signals that are relayed to other parts of the visual system. The human retina has two distinct types of photoreceptors: rods and cones. Rods are very sensitive to light and are responsible for vision from twilight illumination to very dark lighting conditions. Cones are relatively less sensitive and are responsible for vision in daylight to moonlight.

Depending on whether vision is mediated by cones or rods, illumination is broadly divided into *photopic* and *scotopic* ranges. The photopic and scotopic regions

overlap: In the range of illumination between indoor light to moonlight, both rods and cones are active. This region is referred to as the "mesopic range." Rods and cones divide the huge range of illumination into approximately two smaller ranges to which they individually adapt.

The manifestation of adaptation of rods and cones in their respective ranges of illumination is shown in the TVI plots of Figure 7.7. The line toward the lower left corresponds to the thresholds for rods, and the upper right line corresponds to the threshold for cones. In *scotopic* illumination conditions, rods are more sensitive than cones and have a much lower threshold, and the vision in those illumination conditions is mediated by the rod system.

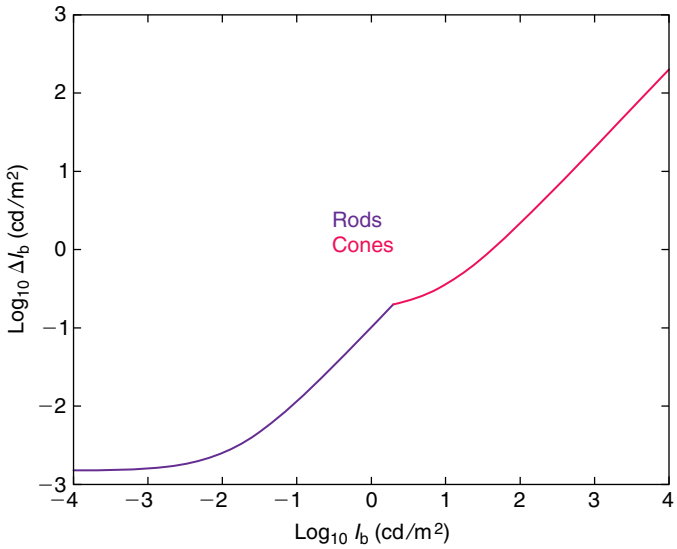


FIGURE 7.7 *TVI curves for rods and cones.*

Starting from dark conditions, as illumination is increased rods become less sensitive and eventually become saturated, and therefore incapable of discriminating contrasts that can be as large as 100:1 [130]. Before this happens, the cone system has taken over, leading to the crossover point seen in Figure 7.7. The equation of the rod curve is given by

$$\Delta I_{\mathrm{b}} = 0.1(I_{\mathrm{b}} + 0.015)$$

and the equation describing the cone curve is

$$\Delta I_{\mathrm{b}} = 0.02(I_{\mathrm{b}} + 8)$$

These equations were obtained by fitting to threshold data [281]. Here, I_{b} is expressed in Trolands (td), a measure for retinal illuminance. A value of 1 td is obtained when a surface with a luminance L of 1 cd/m^2 is viewed through a pupil opening A of 1 mm^2. Thus, retinal illuminance I is given by

$$I = LA,$$

which is measured in candelas (cd).

The diameter d of a circular pupil as a function of background luminance may be estimated [372]. Moon and Spencer [225] propose the following relation between luminance and pupil diameter

$$d = 4.9 - 3\tanh(0.4(\log L + 1.0))$$

Alternatively, de Groot and Gebhard [114] estimate the pupil diameter to be

$$\log d = 0.8558 - 4.01 \times 10^{-4}(\log L + 8.6)^3$$

In both these equations, the diameter d is given in millimeter and L is the luminance in candelas per square meter.

The role of rods and cones in adaptation is important, and deserves consideration when dealing with intensities of very high dynamic range. However, the individual operating ranges of rods and cones are still very large (a million to one). Thus, additional processes must play a significant role in their adaptation.

7.2.3 PHOTOPIGMENT DEPLETION AND REGENERATION

Light is absorbed by the rod and cone photoreceptors through a photochemical reaction. This reaction breaks down photosensitive pigments and temporarily renders them insensitive—a process called "bleaching." The pigments are regenerated by a relatively slow process. Thus, visual adaptation as a function of light intensity could be attributed to the depletion and regeneration of photopigments. Rod photopigments are completely depleted when exposed to light intensities above the mesopic range. It is believed that this depletion renders rods inoperable in the photopic range.

Nonetheless, cone photopigments are not significantly depleted even in bright sunlight, but as evidenced by the TVI relationship, the sensitivity of the cones continues to diminish as a function of background intensity. Therefore, there is a lack of correlation between photopigment concentration and visual sensitivity. This, as well as other experimental evidence, suggests that unless virtually all pigments are bleached, the visual adaptation to different illumination conditions cannot be attributed to photopigment concentration alone [69].

7.2.4 PHOTORECEPTOR MECHANISMS

Electrophysiological studies can be used to detect the response of individual neuronal cells in the HVS. Although the visual system is stimulated with a pattern of light, single-cell recordings are made whereby a thin electrode is held near the cell (extracellular recordings) or inside the cell (intercellular recordings), thus measuring the cell's electrical behavior [243].

Rods and cones convert the absorbed light energy into neural responses, which have been measured with intercellular recordings. Compared with the broad range of background light intensities over which the visual system performs, the photoreceptors respond logarithmically to a rather narrow range of luminances. This range is only about three log units, as shown in Figure 7.8. The log–linear plot in this figure of the luminance-response function is derived from measurements of the response of dark adapted vertebrate rod cells on brief exposures to various luminance values [69].

The response curve of the cones follows the same shape as the response curve of the rod cells. However, because of the higher sensitivity of rod cells to light,

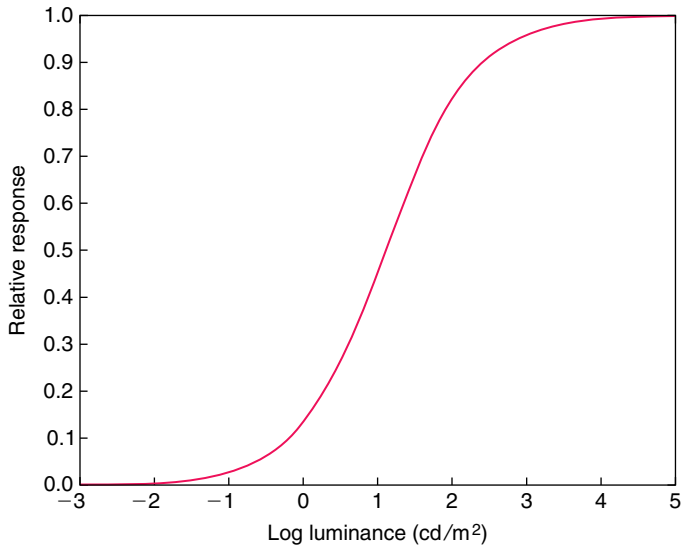


FIGURE 7.8 *The response of dark-adapted vertebrate rod cells to various luminance values. The intensity axis in the image is shown in arbitrary units [69].*

the response curve for the cones is located to the right on the log-luminance axis. Figure 7.9 shows the response curves for both rods and cones.

The response curves for both rods and cones can be fitted with the following equation

$$\frac{R}{R_{\max}} = \frac{I^n}{I^n + \sigma^n}, \tag{7.1}$$

where R is the photoreceptor response ($0 < R < R_{\max}$), $R_{\max}$ is the maximum response, I is light intensity, and σ, the semisaturation constant, is the intensity that causes the half-maximum response. Finally, n is a sensitivity-control exponent, which has a value generally between 0.7 and 1.0 [69].

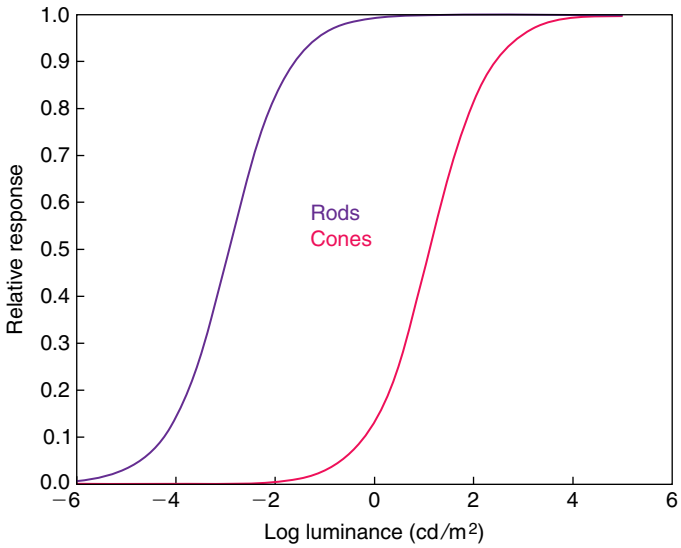


FIGURE 7.9 *The response of dark-adapted rod and cone cells to various intensities in arbitrary units.*

This equation, known as the "Naka–Rushton equation," models an S-shaped function on a log–linear plot and appears repeatedly in both psychophysical experiments [129,2,333,365] and widely diverse, direct neural measurements [234,158, 69,103,111,328]. For $n = 1$, this equation is also known as the "Michaelis–Menten equation." The role of σ in Equation 7.1 is to control the position of the response curve on the horizontal intensity axis. It is thus possible to represent the response curves of rods and cones in Figure 7.9 by simply using two different values of σ, say σ_{rod} and σ_{cone}, in Equation 7.1.

Photoreceptor Adaptation The response curves in Figures 7.8 and 7.9 show that when the dark-adapted photoreceptor is exposed to a brief light of moderately high

intensity, the response reaches its maximum and the photoreceptor is saturated. The photoreceptor loses sensitivity to any additional light intensity. This initial saturation of the photoreceptor matches with our visual experience of blinding brightness when exposed to light at least about 100 times more intense than the current background intensity. But this initial experience does not continue for long. If exposed to this high background intensity for a while, the HVS adapts to this new environment, and we start to function normally again.

Measurements have shown that if photoreceptors are exposed continuously to high background intensities, the initial saturated response does not continue to remain saturated. The response gradually returns toward the dark-adapted resting response, and the photoreceptor's sensitivity to incremental responses is gradually restored. Figure 7.10 shows the downward shift in the measured response at two different background intensities, which are shown as vertical lines.

An interesting observation is that the momentary response never completely returns to the resting response. Rather, it stabilizes on a plateau. Figure 7.10 shows the plateau curve (lower curve) for a range of background intensities. In addition to the restoration of sensitivity, the intensity–response curve measured at any given background intensity is shifted to the right along the horizontal axis, thus ensuring that the narrow response range lies around the background intensity. The shifted curves are shown in Figure 7.11.

Independent measurements have verified that the shapes of the intensity–response curves are independent of the background intensity. However, with background intensity, the position of the response function shifts horizontally along the intensity axis. This shift indicates that, given sufficient time to adapt, the visual system always maintains its log–linear property for approximately three log units of intensity range around any background. This shift is also modeled by the Naka–Rushton equation by simply increasing the value of the semisaturation constant σ as a function of the background intensity. This yields the following modified equation

$$\frac{R}{R_{\max}} = \frac{I^n}{I^n + \sigma_b^n}, \tag{7.2}$$

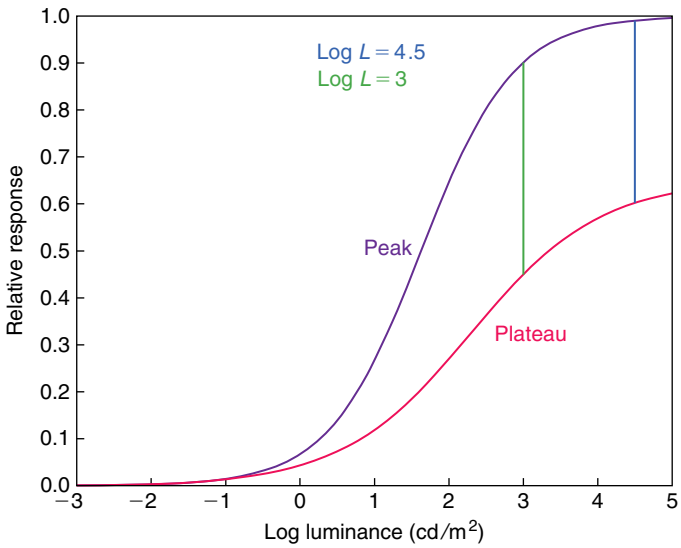


FIGURE 7.10 *Recovery of the response after a long exposure to background intensities [69].*

where the semisaturation σ_b is a function of background intensity I_b. In summary, photoreceptor adaptation, which can be modeled by the Naka–Rushton equation, provides us with the most important mechanism of adaptation.

Response–Threshold Relation The observed relationship between visual threshold and background intensity (Figure 7.5) can be derived from the cellular adaptation model, as shown in Figure 7.12. For this derivation, we assume that the threshold ΔI_b is the incremental intensity required to create an increase in cellular response by a small criterion amount δ [333,112]. Based on this assumption, we derive ΔI_b from the response equation as follows. We begin by rearranging

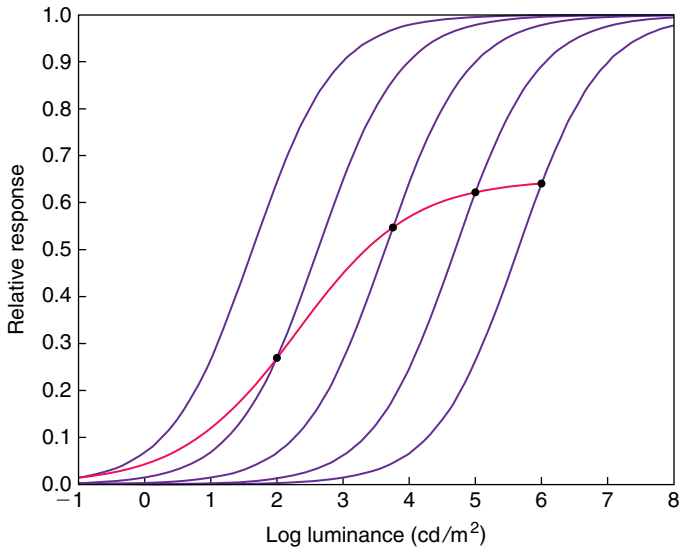
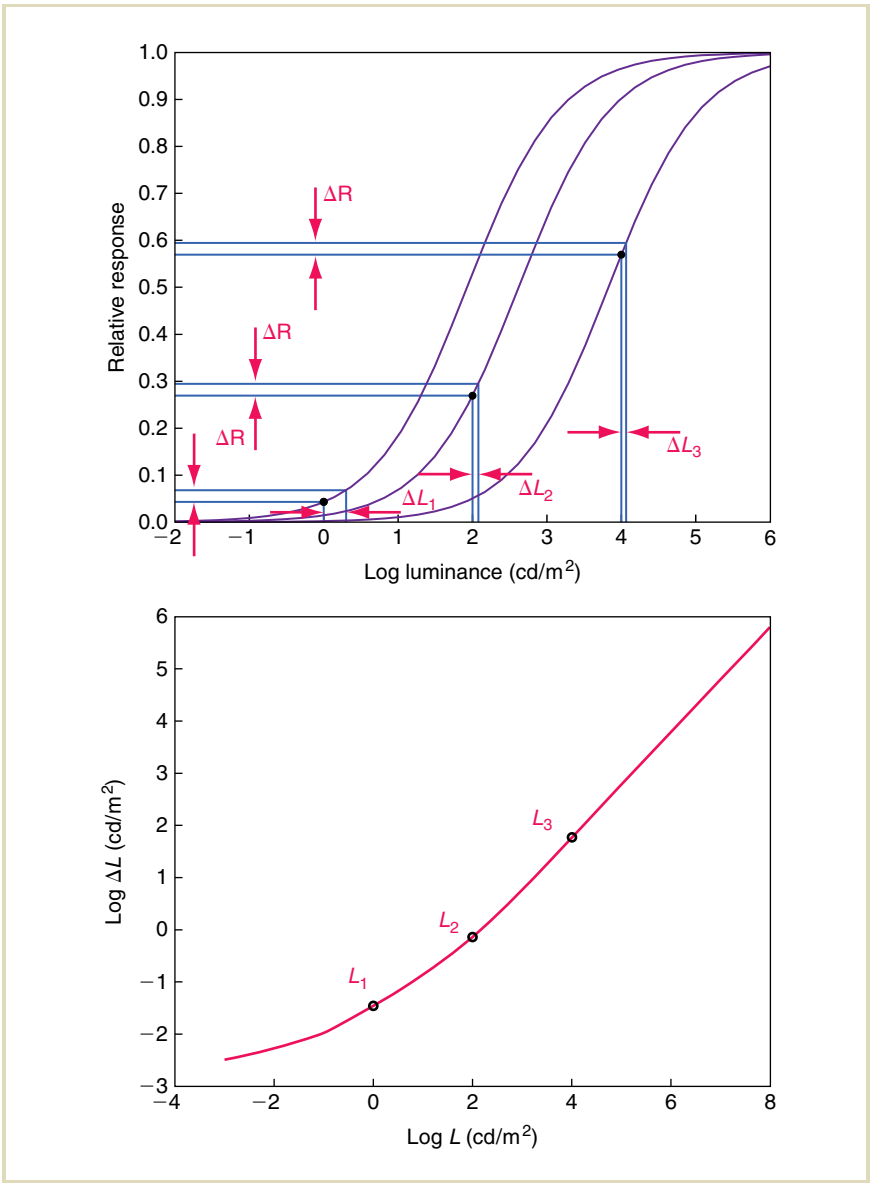


FIGURE 7.11 Photoreceptor response adaptation to different background intensities. The plateau of Figure 7.10 is shown in pink. It represents the locus of the photoreceptor response to the background intensity itself.

FIGURE 7.12 The response–threshold relation. The graph at the top shows photoreceptor response functions at three different background luminances L_i, spaced approximately two log units apart. The response to the background luminance is indicated by the "*" symbol. The ΔL_i's are the luminance increments required to elicit a fixed ΔR response increment. The bottom graph shows the ΔL_i values as function of the background luminances L_i. Note that the curve passing through the ΔL_i values has a shape similar to TVI functions.


Relative response
Log luminance (cd/m^2)
ΔR
ΔR
ΔR
ΔL$_1$
ΔL$_2$
ΔL$_3$
Log ΔL (cd/m^2)
Log L (cd/m^2)
L$_1$
L$_2$
L$_3$

Equation 7.2, yielding

$$I = \sigma_b \left(\frac{R}{R_{max} - R} \right)^{\frac{1}{n}}$$

By differentiating this expression with respect to R, we get

$$\begin{aligned} \frac{dI}{dR} &= \sigma_b \cdot \frac{1}{n} \cdot \left(\frac{R}{R_{max} - R} \right)^{\frac{1}{n}-1} \left(\frac{R_{max}}{(R_{max} - R)^2} \right) \\ &= \sigma_b \cdot \frac{1}{n} \cdot \frac{R_{max}}{(R_{max} - R)^{\frac{n+1}{n}}} R^{\frac{1-n}{n}} \end{aligned}$$

This gives an expression for the incremental intensity, that is, dI required to increase the response of the system by dR. If we assume that the criterion response amount δ for the threshold condition is small enough, then from the above equation, it is possible to compute the expression for ΔI as

$$\begin{aligned} \frac{\Delta I}{\delta} &\cong \frac{dI}{dR} \\ &= \sigma_b \cdot \frac{1}{n} \cdot \frac{R_{max}}{(R_{max} - R)^{\frac{n+1}{n}}} R^{\frac{1-n}{n}} \end{aligned}$$

Note that in all these equations, R is the response of the cellular system exposed to intensity I that may be different from the background intensity I_b to which the system is adapted. For threshold conditions, we can write $R = R_b + \delta$, where R_b is the plateau response of the system at background intensity I_b. Thus,

$$\Delta I = \delta \cdot \sigma_b \cdot \frac{1}{n} \cdot \frac{R_{max}}{(R_{max} - R_b - \delta)^{\frac{n+1}{n}}} (R_b + \delta)^{\frac{1-n}{n}}$$

For dark-adapted cells, the response R_b of the system is 0. The expression of the threshold under dark adaptation is, therefore,

$$\Delta I_{dark} = \delta \cdot \sigma_{dark} \cdot \frac{1}{n} \cdot \frac{R_{max}}{(R_{max} - \delta)^{\frac{n+1}{n}}} \delta^{\frac{1-n}{n}}$$

The relative threshold, $\Delta I / \Delta I_{\text{dark}}$, for adaptation at any other background intensity I_{b}, is

$$\begin{aligned}
\frac{\Delta I}{\Delta I_{\text{dark}}} &= \frac{\sigma_{\text{b}}}{\sigma_{\text{dark}}} \cdot \left(\frac{R_{\text{b}} + \delta}{\delta} \right)^{\frac{1-n}{n}} \left(\frac{R_{\text{max}} - \delta}{R_{\text{max}} - R_{\text{b}} - \delta} \right)^{\frac{n+1}{n}} \\
&\approx \frac{\sigma_{\text{b}}}{\sigma_{\text{dark}}} \cdot \left(\frac{\delta}{R_{\text{b}}} \right)^{\frac{n-1}{n}} \left(\frac{R_{\text{max}}}{R_{\text{max}} - R_{\text{b}}} \right)^{\frac{n+1}{n}} \\
&= \frac{\sigma_{\text{b}}}{\sigma_{\text{dark}}} \cdot \left(\frac{\delta}{R_{\text{max}}} \right)^{\frac{n-1}{n}} \left(\frac{I_{\text{b}}^n + \sigma_{\text{b}}^n}{I_{\text{b}}^n} \right)^{\frac{n-1}{n}} \left(\frac{I_{\text{b}}^n + \sigma_{\text{b}}^n}{\sigma_{\text{b}}^n} \right)^{\frac{n+1}{n}} \\
&= \frac{1}{\sigma_{\text{dark}}} \cdot \left(\frac{\delta}{R_{\text{max}}} \right)^{\frac{n-1}{n}} \frac{\left(I_{\text{b}}^n + \sigma_{\text{b}}^n \right)^2}{I_{\text{b}}^{n-1} \sigma_{\text{b}}^n}
\end{aligned}$$

For $n = 1$ and $I_{\text{b}} = \sigma_{\text{b}}$, $\frac{\Delta I}{\Delta I_{\text{dark}}}$ is directly proportional to I_{b}. This is in agreement with Weber's relation seen in TVI measurements. Thus, Weber's law may be viewed as a behavioral manifestation of photoreceptor adaptation.

The preceding discussion about the various mechanisms of visual adaptation affords the following conclusions:

1. Photoreceptor adaptation plays a very important role in visual adaptation. An appropriate mathematical model of this adaptation (e.g., Equation 7.2) can be effectively used to tone map HDR images. The TVI relation can be derived from the photoreceptor adaptation model and hence can be used as an alternate mathematical model for tone mapping.
2. The combination of rods and cones extends the effective range at which the HVS operates. Depending on the range of intensities present in an image, an appropriate photoreceptor system or combination of them may be chosen to achieve realistic tone mapping.

7.3 VISUAL ADAPTATION MODELS FOR HDR TONE MAPPING

Figure 7.2 outlines a basic framework for HDR tone mapping incorporating models of visual adaptation. The key feature of the framework is that it includes both forward and inverse adaptation models. The forward adaptation model will process scene luminance values and extract visual appearance parameters appropriate for realistic tone mapping. The inverse adaptation model will take the visual appearance parameters and the adaptation parameters appropriate to the display viewing condition and will output the display luminance values.

Either of the visual adaptation models discussed in the previous section, namely the photoreceptor adaptation model or the threshold adaptation model, may be used for forward and inverse adaptation. Most tone-reproduction algorithms available today make use of one of these models. To achieve visually plausible HDR compression, these algorithms use photoreceptor responses or JNDs as correlates of visual appearance. In other words, most tone-reproduction operators aim to preserve either of these quantities. In this section, we discuss the core of various algorithms and show their relation to visual adaptation models. A more detailed discussion of specific tone-reproduction operators is given in Chapter 8.

7.3.1 PHOTORECEPTOR ADAPTATION MODEL FOR TONE MAPPING

This section brings together a significant number of tone-mapping algorithms. They all effectively model photoreceptor adaptation (Equation 7.2), as discussed in Section 7.2. Here, we show the equations implementing specific tone-reproduction operators, and where required rewrite them to demonstrate the similarity with Equation 7.2. In their rewritten form, they are functionally identical to their original forms. It is important to note that although these operators may be derived from the same adaptation equations, they mostly differ in their choice of parameter values. Several of these algorithms include the inverse adaptation step [246, 247,84,83,168,156]), although others omit this step. See also the discussion in Section 7.6.3.

All algorithms that are either directly or indirectly based on photoreceptor adaptation models use Equation 7.2, as it has several desirable properties. First, independent of input intensity, the relative response is limited between 0 and 1. Thus, the relative response output can be directly scaled to displayable pixel values.

Second, by carefully choosing the semisaturation constant, the response function shifts along the intensity axis in such a way that the response of the background intensity is well within the operating range of the response curve.

Finally, the equation has a near-linear response to the intensity in the log domain for about four log units. The intensity ranges of most natural scenes without any highlights or directly visible light sources do not exceed four log units. Thus, such scenes afford an approximately logarithmic relation between intensity and response.

Rational Quantization Function Schlick used the following mapping function for computing display pixel values from pixel intensities (I) [282]:

$$F(I) = \frac{pI}{pI - I + I_{\max}} \qquad \text{Original form}$$

$$= \frac{I}{I + \dfrac{I_{\max} - I}{p}}, \qquad \text{Rewritten form}$$

where $I_{\max}$ is the maximum pixel value and p takes a value in the range $[1,\infty]$.

We can directly relate this equation to Equation 7.2 by setting the exponent n to 1, and substituting $\frac{I_{\max}-I}{p}$ for σ_b. Note that the value of σ_b depends on the value of I itself, which may be interpreted as if the value of every pixel served as the background intensity in the computation of the cellular response.

Gain–Control Function Pattanaik et al. [246] introduced a gain-control function for simulating the response of the HVS and used this gain-controlled response for tone mapping. They proposed two different equations to model the response of the rod and cone photoreceptors:

$$F_{\text{cone}}(I) = \frac{I}{c_1(I_b + c_2)^n},$$

$$F_{\text{rod}}(I) = \frac{r_1}{r_2\left(I_b^2 + r_1\right)^n} \frac{I}{r_3(I_b + r_4)^n},$$

where the constants c_i and r_i are chosen to match certain psychophysical measurements.

In their formulation, I represents the intensity of the image, computed by means of a Gaussian image pyramid. For every level of the pyramid, the background intensity I_b is chosen as the intensity of the pixel at the next-coarser level. These equations have only a superficial resemblance to Equation 7.2 and have been given here for completeness.

S-Shaped Curve Tumblin et al. [325] used an S-shaped curve (sigmoid) as their tone-mapping function:

$$F(I) = \left[\frac{\left(\frac{I}{I_b}\right)^n + \frac{1}{k}}{\left(\frac{I}{I_b}\right)^n + k} \right] \cdot D \qquad \text{Original form}$$

$$= \left[\frac{I^n}{I^n + kI_b^n} + \frac{I_b^n}{k\left(I^n + kI_b^n\right)} \right] \cdot D, \qquad \text{Rewritten form}$$

where k, D, and n are the parameters to adjust the shape and size of the S-shaped curve.

According to the authors, this function is inspired by Schlick's quantization function shown above. The rewritten equation has two parts. The first part is identical to Equation 7.2. The second part of the equation makes it an S-shaped function on a log–log plot.

Photoreceptor Adaptation Model Pattanaik et al. [247,249] and Reinhard and Devlin [272] made explicit use of Equation 7.2 for tone mapping.

Pattanaik et al. used separate equations for rods and cones to account for the intensity in scotopic and photopic lighting conditions. The σ_b values for rods and cones were computed from the background intensity with

$$\sigma_{b,rod} = \frac{c_1\ I_{b,rod}}{c_2 j^2\ I_{b,rod} + c_3(1-j^2)^4\ I_{b,rod}^{1/6}}$$

$$\sigma_{b,cone} = \frac{c_4\ I_{b,cone}}{k^4\ I_{b,cone} + c_5(1-k^4)^2\ I_{b,cone}^{1/3}},$$

where

$$j = \frac{1}{c_6\ I_{b,rod} + 1}$$

$$k = \frac{1}{c_7\ I_{b,cone} + 1}$$

and $I_{b,rod}$ and $I_{b,cone}$ are the background intensities for the rods and cones, respectively.

Reinhard and Devlin provided a much simpler equation for computing σ_b at a given background intensity

$$\sigma_b = (f\ I_b)^m,$$

where f and m are constants and are treated as user parameters in their tone-mapping algorithm.

Photographic Tone-Mapping Function The photographic tone-mapping function used by Reinhard et al. [274,270] bears a strong resemblance to Equation 7.2. The equation can be written in the following form:

$$F(I) = \frac{a\ \dfrac{I}{I_b}}{1 + a\ \dfrac{I}{I_b}} \qquad \text{Original form}$$

$$= \frac{I}{I + \dfrac{I_b}{a}}, \qquad \text{Rewritten form}$$

where a is a scaling constant chosen appropriate to the illumination range (key) of the image scene.

7.3.2 THE TVI MODEL FOR TONE MAPPING

In the previous section, we have shown the relationship between the TVI model and the photoreceptor adaptation model. Thus, it is obvious that the TVI model can be used for tone reproduction.

Ward's tone-mapping algorithm is the first one to make use of the TVI model [343]. In his algorithm, the threshold ΔI_b at any background I_b is used as

a unit to compute a correlate of visual appearance. From the scene pixel luminance I_{scene} and the scene background luminance $I_{\text{b,scene}}$, the ratio

$$k = \frac{I_{\text{scene}} - I_{\text{b,scene}}}{\Delta I_{\text{b,scene}}}$$

is computed. This ratio represents the number of JNDs k by which the pixel differs from the background. Using the display background luminance $I_{\text{b,display}}$ and display adaptation threshold $\Delta I_{\text{b,scene}}$, the model can be inverted to compute display pixel luminances

$$I_{\text{display}} = k\ \Delta I_{\text{b,display}} + I_{\text{b,display}} \tag{7.3}$$

Ferwerda et al. [92] later adapted this concept to compute JNDs specific to rod and cones and used for tone-mapping images with a wide range of intensities. If the background intensity is locally adapted, the log–linear relationship of the threshold to background intensity provides the necessary range compression for HDR images. The issue of local versus global adaptation is discussed in the following section.

7.4 BACKGROUND INTENSITY IN COMPLEX IMAGES

In the earlier sections, we introduced two important adaptation models: the photoreceptor response model and the TVI model. Both of these models require knowledge of the background intensity I_{b}. For any use of either of these models in tone reproduction, I_{b} has to be computed from the intensity of the image pixels. In this section, we describe various methods commonly used to estimate I_{b} from an image.

7.4.1 IMAGE AVERAGE AS I_{b}

The average of the intensity of the image pixels is often used as the value of I_{b}. The average could be the arithmetic average

$$\frac{1}{N}\sum_{i=1}^{N} I_i$$

or geometric average

$$\prod_{i=1}^{N} (I_i + \varepsilon)^{\frac{1}{N}},$$

where N in the equations is the total number of pixels in the image, and ε, an arbitrary small increment, is added to the pixel intensities to take into account the possibility of any zero pixel values in the image. The geometric average can also be computed as

$$\exp\left(\frac{1}{N}\sum_{i=1}^{N}\log(I_i + \varepsilon)\right),$$

where the exponent $\frac{1}{N}\sum_{i=1}^{N}\log(I_i + \varepsilon)$ is the log average of the image pixels.

In the absence of any knowledge of the actual scene, one of these image averages is probably the most appropriate estimate of I_b for most images. A visual adaptation model using such an average is referred to as "global adaptation," and the tone-mapping method is referred to as "global tone mapping." The geometric average is often the preferred method of average computation. This is mostly because the computed background intensity is less biased toward outliers in the image, and the relationship between intensity and response is log–linear.

7.4.2 LOCAL AVERAGE AS I_b

In images with a very HDR, the intensity change from region to region can be significant. Hence the image average, also called "global average," is not a good representative of the background intensity of the whole image. The proper approach in such cases would be to segment the image into regions of low dynamic range (LDR) and use the average of pixels in each region. Although segmentation for the purpose of tone reproduction can be a difficult task [374], plausible results on the basis of segmentation have been demonstrated [161,163,165] (see Section 8.3.4).

An alternate and popular approach is to compute a local average for every pixel p in the image from its neighboring pixels. The various techniques under this category

include box filtering and Gaussian filtering. These techniques are simple to compute:

$$I_{b,p} = \frac{1}{\sum_{i \in \Omega} w(p,i)} \sum_{i \in \Omega} w(p,i)\, I_i \tag{7.4}$$

For Gaussian filtering,

$$w(p,i) = \exp\left(-\frac{\|p-i\|^2}{s^2}\right)$$

and for box filtering,

$$w(p,i) = \begin{cases} 1 & \text{for } \|p-i\| < s, \\ 0 & \text{otherwise.} \end{cases}$$

In the above equations, Ω represents all the pixels of the image around p, $\|.\|$ is the spatial distance function, and s is a user-defined size parameter. Effectively, the value of s represents the size of a circular neighborhood around pixel p that influences the average value.

Although for most pixels in the image the local average computed in this fashion is representative of the background intensity, the technique breaks down at HDR boundaries. This is because the relatively large disparity in pixel intensities in the neighborhood of the boundary biases the average computation. Thus, the background intensity computed for pixels on the darker side of the boundary is positively biased and that computed for the pixels on the brighter side is negatively biased. This biasing gives rise to halo artifacts in the tone-mapped images. Figure 7.13 highlights the problem. The image shown is computed using local box-filtered values for the background intensity. Note the dark band on the darker side of the intensity boundary. Although not noticeable, similar bright banding exists on the brighter side of the boundary.

This problem can be avoided by computing the average only from nearby pixels whose intensities are within a reasonable distance from the pixel under consideration. The tone-mapped image in Figure 7.13 shows a result obtained using such an adaptive computational approach. There is a significant improvement in visible contrast, but at an increased cost of computation. Given below are two such computational approaches.

FIGURE 7.13 *Halo artifacts associated with the use of I_b computed by local averaging. The artifacts are most noticeable at illumination boundaries.*

Local Average Using Variable Size Neighborhood In this approach, the size parameter s in Equation 7.4 is adaptively varied. Reinhard et al. and Ashikhmin simultaneously proposed this very simple algorithm [274,12]. Starting from a value of s equal to 1, they iteratively double its value until the pixels from across the HDR boundary start to bias the average value. They assume that the average is biased if it differs from the average computed with the previous size by a tolerance amount. They use this s in Equation 7.4 for computing their local average.

Local Average Using Bilateral Filtering In this approach, the size parameter s remains unchanged, but the pixels around p are used in the average summation only if their intensity values are similar to the intensity of p. The similarity can be user-defined. For example, the intensities may be considered similar if the difference or the ratio of the intensities is less than a predefined amount. Such an approach may be implemented by filtering both in spatial and intensity domains. The name "bilateral" derives from this dual filtering

$$I_{b,p} = \frac{1}{\sum_{i\in\Omega} w(p,i)g(I_p, I_i)} \sum_{i\in\Omega} w(p,i)g(I_p, I_i)I_i \tag{7.5}$$

FIGURE 7.14 *Tone mapping using adaptive local average computation.*

where $w(p, i)$ and $g(I_p, I_i)$ are the two weighting functions to take into account the dual proximity. The forms of these weighting functions can be similar except that their parameters are different: for $g(I_p, I_i)$ the parameters are the intensities of the two pixels, and for $w(p, i)$ the parameters are the positions of the two pixels.

Durand and Dorsey use Gaussian functions for both domains [74]. Pattanaik and Yee use a circular box function for $w(p, i)$, and an exponential function for $g(I_p, I_i)$ [249]. Choudhury and Tumblin have proposed an extension to this technique to account for gradients in the neighborhood. They named their extension "trilateral filtering" [40]. Finally, very efficient implementations of bilateral filtering have appeared that dramatically reduce the computational cost of this approach [357,38,244]. An example of tone mapping whereby the average is derived locally is shown in Figure 7.14.

Figure 7.15 shows a linearly scaled version of an original HDR image and images tone mapped using several of the adaptive local adaptation techniques discussed in this section.

7.4.3 MULTISCALE ADAPTATION

Although the use of local averages as the background intensity is intuitive, the choice of the size of the locality is mostly ad hoc. In this section, we provide some empirical

(a)

(b)

(c)

(d)

(e)

FIGURE 7.15 *Local averages for a sample HDR image shown in (a). The images (b) and (c) were computed using Equation 7.4, and images (d) and (e) were computed using Equation 7.5 . Equal weighting is used for images (b) and (d) and Gaussian weighting for images (c) and (e). $g(I_p, I_i)$ for figure (d) is from Pattanaik [249] and for figure (e) from Durand and Dorsey [74]. (HDR image courtesy Columbia University)*

support for the use of local averages and the associated importance to the size of the locality.

Physiological and psychophysical evidence indicates that the early stages of visual processing can be described as the filtering of the retinal image by band-pass mechanisms sensitive to patterns of different scales [364]. These band-pass mechanisms adapt independently to the average intensity within a region of a scene defined by their spatial scale. In a complex scene, this average will be different at different scales, so the mechanisms will all be in different states of adaptation.

Thus, to correctly account for the changes in vision that occur with changes in the level of illumination, we need to consider local adaptation at different spatial scales within HDR environments. Peli suggests that an appropriate way to characterize the effects of local adaptation on the perception of scenes is to use low-pass images that represent the average local luminance at each location in the image at different spatial scales [250]. Reinhard et al. [274] and Ashikmin [12] use this multiscale approach to adaptively select the effective neighborhood size. Other multiscale adaptations also demonstrate the usefulness of the multiscale nature of the visual system in HDR tone mapping [246,186] (see also Sections 8.2.1 and 8.3.5).

7.5 DYNAMICS OF VISUAL ADAPTATION

In earlier sections, we discussed the adaptation of the visual system to background intensity. However, visual adaptation is not instantaneous. In the course of the day, light gradually changes from dim light at dawn to bright light at noon, and back to dim light at dusk. This gradual change gives the visual system enough time to adapt and hence the relatively slow nature of visual adaptation is not noticed. However, any sudden and drastic change in illumination, either from light to dark or dark to light, makes the visual system momentarily lose its normal functionality. This loss of sensitivity is experienced as total darkness during a light-to-dark transition, and as a blinding flash during a dark-to-light transition. Following this momentary loss in sensitivity, the visual system gradually adapts to the prevailing illumination and recovers its sensitivity. This adaptation is also experienced as a gradual change in perceived brightness of the scene.

The time course of adaptation—the duration over which the visual system gradually adapts—is not symmetrical. Adaptation from dark to light, known as *light adaptation*, happens in a matter of seconds, whereas *dark adaptation*, adaptation from light to dark, occurs over several minutes. We often experience dark adaptation phenomena when we enter a dim movie theater for a *matinee*. Both adaptation phenomena are experienced when we drive into and out of a tunnel on a sunny day. The capability of capturing the full range of light intensities in HDR images and video poses new challenges in terms of realistic tone mapping of video image frames during the time course of adaptation.

In Section 7.2, we argued that vision is initiated by the photochemical interaction of photons with the photopigments of the receptor. This interaction leads to bleaching and hence the loss of the photopigments from the receptors. The rate of photon interaction and hence the rate of the loss in photopigments is dependent on the intensity of light, on the amount of photopigment present, and on photosensitivity. A slow chemical regeneration process replenishes the lost photopigments. The rate of regeneration depends on the proportion of the bleached photopigments and on the time constant of the chemical reaction.

From the rate of bleaching and the rate of regeneration, it is possible to compute the equilibrium photopigment concentration for a given illumination level. Because the rate of photon interaction is dependent on the amount of photopigments present and because the bleaching and regeneration of bleached photopigments are not instantaneous, visual adaptation and its time course were initially thought to be directly mediated by the concentration of the unbleached photopigments present in the receptor.

Direct cellular measurements on isolated and whole rat retinas by Dowling ([69], Chapter 7) show that dark adaptation in both rods and cones begins with a rapid decrease in threshold followed by a slower decrease toward the dark adaptation threshold. The latter slow adaptation is directly predicted by photopigment concentrations, whereas the rapid adaptation is attributed almost entirely to a not-well-understood but fast neural adaptation process.

The Naka–Rushton equation (Equation 7.1) models the photoreceptor response and accounts for visual adaptation by changing the σ_b value as a function of background intensity. Photoreceptor adaptation and pigment bleaching have been proposed to account for this change in σ_b value. Valeton and van Norren have modeled

the contribution of these two mechanisms to the increase in σ_b by

$$\sigma_b = \sigma_{dark}\ \sigma_{b,neural}\ \sigma_{b,bleach}, \tag{7.6}$$

where σ_{dark} is the semisaturation constant for dark conditions, $\sigma_{b,neural}$ accounts for the loss in sensitivity because of the neural adaptation, and $\sigma_{b,bleach}$ accounts for the loss in sensitivity because of the loss of photopigment [328]. The value of $\sigma_{b,bleach}$ is inversely proportional to the fraction of unbleached photopigments at the background light.

Pattanaik et al. [247] extended the use of the adaptation model to compute the time course of adaptation for simulating visual effects associated with a sudden change in intensities from dark to light or vice versa [247]. They use a combination of Equations 7.1 and 7.6 to carry out the simulation,

$$\frac{R}{R_{max}} = \frac{I^n}{I^n + \sigma_b^n(t)}$$

$$\sigma_b(t) = \sigma_{dark}\ \sigma_{b,neural}(t)\ \sigma_{b,bleach}(t),$$

where time-dependent changes of $\sigma_{b,neural}$ and $\sigma_{b,bleach}$ are modeled with exponential decay functions.

7.6 DESIGN CONSIDERATIONS

As all tone-reproduction operators are aimed at more or less the same problem, namely the appropriate reduction of dynamic range for the purpose of display, there are several ideas and concepts that are shared by many of them. The functional form of a significant group of operators is discussed in the preceding sections. In addition, the input image is often expected to be calibrated in real-world values. Further, color is treated similarly by many operators. At the same time, several operators apply compression in logarithmic space, whereas others compress in linear space. Finally, most local operators make use of suitably blurred versions of the input image. Each of these issues is discussed in the following sections.

7.6.1 CALIBRATION

Several tone-reproduction operators are inspired by aspects of human vision. The human visual response to light at different levels is nonlinear, and photopic and scotopic lighting conditions in particular lead to very different visual sensations, as discussed in the preceding chapter. For those tone-reproduction operators, it is important that the values to be tone mapped are specified in real-world units, that is, in candelas per square meter. This allows operators to differentiate between a bright, day-lit scene and a dim night scene. This is not generally possible if the image is given in arbitrary units; see, for example, Figure 7.16.

However, unless image acquisition is carefully calibrated, images in practice may be given in arbitrary units. For several tone-reproduction operators, this implies

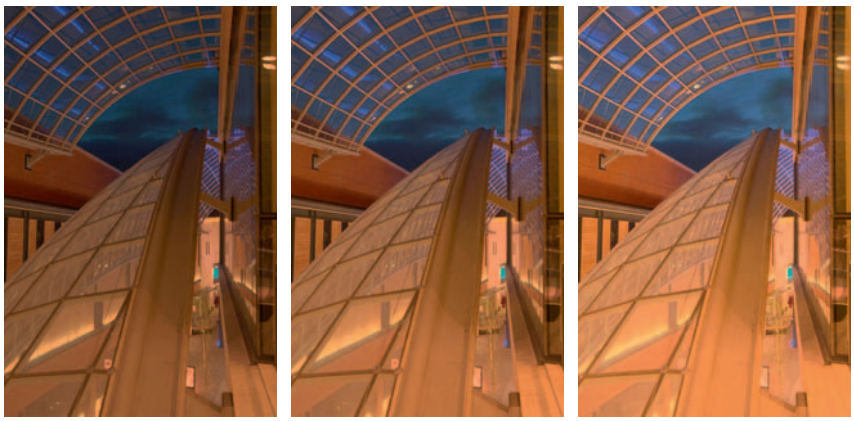


FIGURE 7.16 *This image is given in arbitrary units, and it is tone mapped three times with different parameters, using the photographic operator discussed in Section 8.1.2. Without knowing the actual scene, it is difficult to assess which of these three renditions is most faithful to the actual scene. If the data were properly calibrated, the absolute values in the image would allow the overall level to be determined. (Image courtesy of Tania Pouli)*

that, for instance, an uncalibrated night image may be tone mapped as if it were a representation of a day-lit scene. Displaying such an image would give a wrong impression.

Images may be calibrated by applying a suitably chosen scale factor. Without any further information, the value of such a scale factor can realistically only be approximated either by trial and error, or by making further assumptions on the nature of the scene. In this chapter and the next, we show a progression of images for each operator requiring calibrated data. These images are generated with different scale factors such that the operator's behavior on uncalibrated data becomes clear. This should facilitate the choice of scale factors for other images.

Alternatively, it is possible to use heuristics to infer the lighting conditions for scenes depicted by uncalibrated images. In particular, the histogram of an image may reveal if an image is overall light or dark, irrespective of the actual values in the image. Figure 7.17 shows histograms of dark, medium, and light scenes. For many natural scenes, a dark image will have pixels with values that are located predominantly toward the left of the histogram. A light image will often display a peak toward the right of the histogram, with images in between having a peak somewhere in the middle of the histogram.

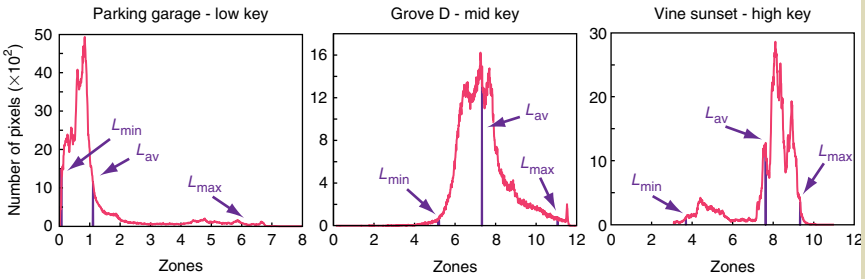


FIGURE 7.17 *Histograms for scenes that are overall dark (left), medium (middle), and light (right).*

An important observation is that the shape of the histogram is determined both by the scene being captured and the capture technique used. As our main tool for capturing HDR images uses a limited set of differently exposed LDR images (discussed in Chapter 5), images with the sun directly visible will still contain burned-out regions. Similarly, low-level details in nighttime scenes may also not be represented well in an HDR image captured with this method. These limitations affect the shape of the histogram and, therefore, the estimation of the key of the scene.

A number that correlates to the peak found in a histogram, but is not equal to the location of the peak, is the log average luminance found in the image:

$$L_{\text{av}} = \exp\left(\frac{1}{N}\sum_{x,y}\log(L_{\text{W}}(x, y))\right), \tag{7.7}$$

where the summation is only over nonzero pixels.

The key of a scene, a unitless number that relates to the overall light level, may be inferred from a histogram. It is, thus, possible to relate empirically the log average luminance to the minimum and maximum luminance in the histogram (all three are shown in the histograms of Figure 7.17). The key α may be estimated with [270]:

$$f = \left(\frac{2\log_2 L_{\text{av}} - \log_2 L_{\text{min}} - \log_2 L_{\text{max}}}{\log_2 L_{\text{max}} - \log_2 L_{\text{min}}}\right) \tag{7.8}$$

$$\alpha = 0.18 \times 4^f \tag{7.9}$$

Here, the exponent f computes the distance of the log average luminance to the minimum luminance in the image relative to the difference between the minimum and maximum luminance in the image. To make this heuristic less dependent on outliers, the computation of the minimum and maximum luminance should exclude around 1% of the lightest and darkest pixels.

For the photographic tone-reproduction operator (discussed in Section 8.1.2), a sensible approach is to first scale the input data such that the log average luminance

is mapped to the estimated key of the scene:

$$L'_{\mathrm{W}}(x, y) = \frac{\alpha}{L_{\mathrm{av}}} L_{\mathrm{W}}(x, y) \tag{7.10}$$

Although unproven, this heuristic may also be applicable to other tone-reproduction techniques that require calibrated data. However, in any case, the best approach would be always to use calibrated images.

7.6.2 COLOR IMAGES

The HVS is a complex mechanism with several idiosyncrasies that need to be accounted for when preparing an image for display. Most tone-reproduction operators attempt to reduce an image in dynamic range while keeping the response of HVS to the reduced set of intensities as constant. This leads to various approaches that aim at preserving brightness, contrast, appearance, and/or visibility.

However, it is common practice among many tone-reproduction operators to exclude a comprehensive treatment of color. With few exceptions, it is generally accepted that dynamic range compression should be executed on a single luminance channel. Although this is the current state of affairs, this may change in the near future; the field of color appearance modeling and tone reproduction are growing closer together, as seen in Pattanaik's multiscale observer model [246] as well as more recent developments such as Johnson and Fairchild's iCAM model [84,83] and Reinhard's photoreceptor-based operator [272].

Most other operators derive a luminance channel from the input RGB values, as shown in Section 2.4, and then compress the luminance channel. The luminance values computed from the input image are called "world luminance" (L_{w}). The tone-reproduction operator of choice will take these luminance values and produce a new set of luminance values L_{d}. The subscript d indicates "display" luminance. After compression, the luminance channel needs to be recombined with the uncompressed color values to form the final tone-mapped color image.

To recombine luminance values into a color image, color shifts will be kept to a minimum if the ratio between the color channels before and after compression is kept constant [120,300,282]. This may be achieved if the compressed image

$R_d G_d B_d$ is computed as follows:

$$\begin{bmatrix} R_d \\ G_d \\ B_d \end{bmatrix} = \begin{bmatrix} L_d \dfrac{R_W}{L_W} \\ L_d \dfrac{G_W}{L_W} \\ L_d \dfrac{B_W}{L_W} \end{bmatrix}$$

Should there be a need to exert control over the amount of saturation in the image, the fraction in the above equations may be fitted with an exponent s, resulting in a per-channel gamma correction

$$\begin{bmatrix} R_d \\ G_d \\ B_d \end{bmatrix} = \begin{bmatrix} L_d \left(\dfrac{R_W}{L_W} \right)^s \\ L_d \left(\dfrac{G_W}{L_W} \right)^s \\ L_d \left(\dfrac{B_W}{L_W} \right)^s \end{bmatrix}$$

The exponent s is then given as a user parameter, which takes values between 0 and 1. For a value of 1, this method defaults to the standard method of keeping color ratios constant. For smaller values, the image will appear more desaturated, as shown in Figure 7.18, which demonstrates the effect of varying the saturation control parameter s. Full saturation is achieved for a value of $s = 1$. Progressively, more desaturated images may be obtained by reducing this value.

An alternative and equivalent way to keep the ratios between color channels constant is to convert the image to a color space that has a luminance channel and two chromatic channels such as the Yxy color space. If the image is converted to Yxy space first, the tone-reproduction operator will compress the luminance channel Y and the result is converted back to RGB values for display. This approach is functionally equivalent to preserving color ratios.

FIGURE 7.18 *The saturation parameter s is set to 0.6, 0.8, and 1.0 (in reading order).*

7.6.3 FORWARD AND BACKWARD MODELS

Many tone-reproduction operators are modeled after some aspects of human vision. The computed display values, therefore, essentially represent perceived quantities. If we assume that the model is an accurate representation of some aspect of the HVS, displaying the image and observing it will cause the HVS to interpret these perceived values as luminance values.

The HVS thus applies a second perceptual transform on top of the one applied by the algorithm. This is formally incorrect [273]. A good tone-reproduction operator should follow the same common practice as used in color appearance modeling, and apply both a forward and an inverse transform [82], as argued in Section 7.3. The forward transform can be any algorithm thought to be effective at compressing luminance values. The inverse transform will then apply the algorithm in reverse, but with display parameters inserted. This approach compresses luminance values into perceived values, whereas the inverse algorithm will convert the perceived values back into luminance values.

Tumblin and Rushmeier's approach correctly takes this approach, as do the multiscale observer model [246], all color appearance models, and gradient-domain operators, as well as some other tone-reproduction operators. However, several perceptually based operators are applied only in forward mode. Although these operators are known to produce visually plausible results, we note that they are effectively not producing display luminances, but rather are producing brightness or other equivalent perceptual attributes.

Here, we discuss the implications of adding an inverse step to sigmoidal compression. Recall that Equation 7.1 can be rewritten as

$$V(x, y) = \frac{L_v^n(x, y)}{L_v^n(x, y) + g^n(x, y)}, \tag{7.11}$$

where V is a perceived value (for instance a voltage, if this equation is thought of as a simple model of photoreceptor physiology). The function g returns either as a globally or locally computed adaptation value, which is based on the image values.

To convert these perceived values back to luminance values, this equation would need to be inverted, whereby g is replaced with a display-adaptation value. For instance, we could try to replace g with the mean display luminance, $L_{d,\text{mean}}$. The other user parameter in this model is the exponent n, that for the inverse model we will replace with a display-related exponent m. By making these substitutions, we have replaced all the image-related user parameters (n and g) with their display-related equivalents (m and $L_{d,\text{mean}}$). The resulting inverse equation, computing display values L_d from previously computed perceived values V, is then,

$$L_d(x, y) = \left(\frac{V(x, y)\, L_{d,\text{mean}}^m}{1 - V(x, y)} \right)^{1/m} \tag{7.12}$$

For a conventional display, we would set $L_{d,\text{mean}}$ to 128. The exponent m is also a display-related parameter and determines how display values are spread around the mean display luminance. For LDR display devices, this value can be set to 1, thereby simplifying the above equation to

$$L_d(x, y) = \frac{V(x, y)\, L_{d,\text{mean}}}{1 - V(x, y)} \tag{7.13}$$

The computation of display values is now driven entirely by the mean luminance of the image (through the computation of g), the mean display luminance $L_{\text{d,mean}}$, as well as the exponent n, which specifies how large a range of values around the mean image luminance will be visualized. As a result, the inverse transform may create display values that are outside the display range. These will have to be clamped.

As for most display devices, the peak luminance $L_{d,\text{max}}$ as well as the black level $L_{d,\text{min}}$ are known, and it is attractive to use these natural boundaries to clamp the display values against.

Given that we will gamma correct the image afterward, we may assume that the display range of L_d is linear. As such, we can now compute the mean display luminance as the average of the display's black level and peak luminance:

$$L_{d,\text{mean}} = \frac{1}{2}\left(L_{d,\text{min}} + L_{d,\text{max}}\right) \tag{7.14}$$

As a result, all display-related parameters are now fixed, leaving only the image-dependent parameters n as well as the choice of semisaturation constant $g(I_p, I_i)$. For our example, we will follow Reinhard et al. and set $g(I_p, I_i) = \bar{L}_v/k$, where the user parameter k determines how overall light or dark the image should be reproduced (see Section 8.1.2) [274]. The exponent n can be thought of as a measure of how much contrast there is in the image.

Two results using visually determined optimal parameter settings are shown in Figure 7.19. The display settings are those for an average display with an assumed black level of 1cd/m^2 and a peak luminance of $300\,\text{cd/m}^2$. As a consequence, $L_{d,\text{mean}}$ was set to 150.5. The left image is reproduced in a satisfactory manner, while there is too much burn-out in the window of the right image. Here, it is difficult to find a good trade-off between having less burn-out and introducing other artifacts with this method.

FIGURE 7.19 *Forward and inverse model with $n = 0.7$ and $k = 0.3$ for an HDR image with a relatively modest dynamic range of 2.8 log units (left) and an image with a much higher dynamic range (right); $n = 0.7$, $k = 0.08$, 8.3 log units dynamic range.*

For comparison, we show the forward-only transform with otherwise identical parameter settings in Figure 7.20. Note that the left image looks more flat now, largely because the exponent n is no longer optimal. The window in the right image now appears more correct, as the brown glass panels are now clearly visible. We also show the output of the photographic operator for the same pair of images in Figure 7.21. The exponent n is effectively set to 1, but the key value is the same as in the previous figures. Although these images are computed with a forward transform only, their visual appearance remains closer to the real environment than the images in Figure 7.19.

Finally, it is desirable that a tone-reproduction operator does not alter an image that is already within the display range [66]. In the model proposed, here, this is implicitly achieved; as for $n = m$ and $g = L_{d,\text{mean}}$, the inverse transform is the true inverse of the forward transform.

Although applying both a forward and an inverse transform is formally the correct approach to tone reproduction, there is a problem for images with a very HDR. For such images it is difficult, if not impossible, to find parameter settings that lead to an acceptable compression. To see why this is, we can plug the forward transform

FIGURE 7.20 Forward model only with $n = 0.7$, $k = 0.3$ (left) and $n = 0.7$, $k = 0.08$ (right).

FIGURE 7.21 Photographic operator with key $k = 0.3$ (left) and $k = 0.08$ (right). By using the photographic operator, we have effectively changed the exponent to $n = 1$.

into the inverse:

$$L_d = \left(\frac{\dfrac{L^n}{L^n + \left(\bar{L}/k\right)^n} L_{d,\text{mean}}^m}{1 - \dfrac{L^n}{L^n + \left(\bar{L}/k\right)^n}} \right)^{1/m} \tag{7.15}$$

$$= \frac{L^{n/m} L_{d,\text{mean}}}{\left(\bar{L}/k\right)^{n/m}} \tag{7.16}$$

$$= c\, L^{n/m}, \tag{7.17}$$

where

$$c = \frac{L_{d,\text{mean}}}{\left(\bar{L}/k\right)^{n/m}} \tag{7.18}$$

is a constant. Of course, this is essentially the same result as was obtained by matching image and display brightness in Tumblin and Rushmeier's brightness-matching operator. Hence, applying a sigmoid in forward and inverse mode amounts to applying a power function.

In our experience, this approach works very well in cases where a medium amount of compression is required. For instance, a medium dynamic range image can be effectively tone mapped for display on an LDR display device. Alternatively, it should be possible to tone map most HDR images for display on HDR display devices using this technique. However, for high-compression ratios, a different approach would be required.

A direct consequence is that we predict that color appearance models such as CIECAM02 cannot be extended to transform data over large ranges. It is well-known that CIECAM02 was never intended for transforming between significantly different display conditions. However, this can be attributed to the fact that the psychophysical data on which this model is based was gathered over a limited dynamic range. The above findings suggest that, in addition, extension of CIECAM02 to accommodate large compression ratios would require a different functional form.

Whether the inclusion of spatial processing, such as a spatially varying semisaturation constant, yields more satisfactory results remains to be seen. As can be understood from Equation 7.16, replacing $g(I_p, I_i) = \bar{L}_v/k$ with a spatially varying function means that each pixel is divided by a spatially determined denominator. Such an approach was pioneered in Chiu et al.'s early work [39] and has been shown to be prone to haloing artifacts. To minimize the occurrence of halos in such a scheme, the size of the averaging kernel used to compute $g(I_p, I_i)$ must be chosen to be very large—typically, a substantial fraction of the whole image. But in the limit that the filter kernel becomes the whole image, this means that each pixel is divided by the same value, resulting in a spatially invariant operator.

7.7 SUMMARY

In this chapter, we propose the view that modeling of human visual adaptation is key to realistic tone mapping of HDR images. We present photoreceptor adaptation as the most important factor responsible for visual adaptation and present Equation 7.1 as the mathematical model for this adaptation. The relation between various tone-mapping algorithms and the photoreceptor adaptation model is made evident. Background intensity is a key component in this model. Some of the commonly used methods for computing this background intensity in images are discussed. We conclude by showing the usefulness of a human visual model in realistic simulation of visual effects associated with the wide range of real-life illuminations.

While this chapter shows the similarities between several current tone-reproduction operators, the following chapter discusses their differences and present each tone-reproduction operator in detail.

Tone-Reproduction Operators

08

While the previous chapter presented a tutorial on various aspects of tone reproduction, and showed that a large class of operators are at the high level equivalent, at the implementation level there still exist significant differences which ought to be taken into account when implementing specific tone-reproduction operators. Without aiming for completeness, in this chapter, we discuss the most prevalent tone-reproduction operators, ranging from sigmoidal compression to image appearance models to a collection of perception- and engineering-based methods.

8.1 SIGMOIDAL TONE-REPRODUCTION OPERATORS

As discussed in this chapter, a relatively well-populated class of operators uses an S-shaped curve (usually when plotted in a log–linear diagram). Sigmoidal functions are those that have a single inflection point. For tone reproduction, sigmoids tend to also produce output in the $[0,1]$ range, so that their display range is naturally limited without the need for additional clamping. A collection of the most prevalent sigmoidal algorithms is discussed in this section.

8.1.1 PHOTORECEPTOR MODEL

Logarithmic compression may be viewed as effectively computing a density image. The output, therefore, resembles the information stored in a negative. Although

© 2010, Elsevier Inc. All rights reserved.
DOI: 10.1016/B978-0-12-374914-7.00008-7

this metaphor holds, logarithmic responses are also sometimes attributed to parts of the human visual system (HVS). This is intrinsically incorrect, although the HVS responds approximately logarithmically over *some* of its operating range (see Section 7.2). Cells in the HVS communicate with impulse trains, where the frequency of these impulse trains carries the information. Notable exceptions are the first few layers of cells in the retina, which communicate by generating graded potentials. In any case, this physiological substrate does not enable communication of negative numbers. The impulse frequency may become zero, but there is no such thing as negative frequencies. There is also an upper bound to realizable impulse frequencies.

Logarithms, however, may produce negative numbers. For large input values, the output may become arbitrarily large. At the same time, over a range of values the HVS may produce signals that appear to be logarithmic. Outside this range, responses are no longer logarithmic but tail off instead. A class of functions that approximates this behavior reasonably well are sigmoids, or S-shaped functions, as discussed in Chapter 7. When plotted on a log–linear graph, the middle portion of such sigmoids are nearly linear and thus resemble logarithmic behavior. Moreover, sigmoidal functions have two asymptotes, one for very small values and one for large values.

This gives sigmoidal functions the right mathematical properties to be a possible candidate for modeling aspects of the HVS. Evidence from electrophysiology confirms that photoreceptors of various species produce output voltages as function of light intensity received that may be accurately modeled by sigmoids.

Naka and Rushton were the first to measure photoreceptor responses and managed to fit a sigmoidal function to their data [234]. For the purpose of tone reproduction, the formulation by Hood et al. is practical [131]:

$$V(x, y) = \frac{I(x, y)}{I(x, y) + \sigma I_{\mathrm{a}}(x, y)}$$

Here, I is the photoreceptor input, V is the photoreceptor response, and σ is the semisaturation constant, which is a function of the receptor's adaptation level I_{a}. The semisaturation constant thus determines to which value of V the adaptation level is mapped and so provides the flexibility needed to tailor the curve to the image that is being tone mapped. For practical purposes, the semisaturation constant may be

computed from the adaptation value I_a as follows:

$$\sigma(I_a(x, y)) = (f\, I_a(x, y))^m$$

In this equation, f and m are user parameters that need to be specified on a per-image basis. The scale factor f may be used to steer the overall luminance of the tone-mapped image and can initially be estimated to be 1. Images created with different values of f are shown in Figure 8.1.

FIGURE 8.1 *Luminance control with user parameter f in Reinhard and Devlin's photoreceptor-based operator. User parameter f, set here to $\exp(-8)$, $\exp(0) = 1$, and $\exp(8)$. The top right image shows the default value.*

It should be noted that in electrophysiological studies, the exponent m also features and tends to lie between 0.2 and 0.9 [131]. A reasonable initial estimate for m may be derived from image measures such as the minimum, maximum, and average luminance:

$$m = 0.3 + 0.7\,k^{1.4}$$

$$k = \frac{L_{\max} - L_{\text{av}}}{L_{\max} - L_{\min}}$$

The parameter k may be interpreted as the key of the image, that is, it is a measure of how light or dark the image is on average. The nonlinear mapping from k to exponent m is determined empirically. The exponent m is used to steer the overall impression of contrast, as shown in Figure 8.2.

A tone-reproduction operator may be created by equating display values to the photoreceptor output V, as demonstrated by Reinhard and Devlin [272]. Note that this operator is applied to each of the red, green, and blue color channels separately. This is similar to photoreceptor behavior where each of the three different cone types is thought to operate largely independently. Also note that sigmoidal functions that are part of several color appearance models, such as the Hunt model [134], CIECAM97 [133], and CIECAM02 [226] (see Section 2.9), are executed independently of the red, green, and blue channels. This approach may account for the Hunt effect, which predicts the desaturation of colors for both light and dark pixels, but not for pixels with intermediate luminances [134].

The adaptation level I_{a} may be computed in traditional fashion, for instance, as the (log) average luminance of the image. However, additional interesting features such as light and chromatic adaptation may be modeled by a slightly more elaborate computation of I_{a}.

Strong color casts may be removed by interpolating between the luminance value $L(x, y)$ of the pixel and the red, green, and blue values of each pixel $I_{\text{r|g|b}}(x, y)$. This produces a different adaptation level for each pixel individually, which is controlled by a user-specified interpolation weight c:

$$I_{\text{a}}(x, y) = c\,I_{\text{r|g|b}}(x, y) + (1 - c)\,L(x, y)$$

FIGURE 8.2 *The exponent m in Reinhard and Devlin's operator. Images are shown with exponent m set to 0.2, 0.4, 0.6, and 0.8 (in reading order).*

This approach achieves von Kries–style color correction by setting $c = 1$, whereas no color correction is applied if c equals 0. We also call this color adjustment "chromatic adaptation," and its effect is shown in Figure 8.3 for three values of c.

Similarly, the adaptation level may be thought of as determined by the current light level to which a receptor is exposed, as well as levels to which the receptor was exposed in the recent past. Because the eye makes saccadic movements, and also because there is the possibility of lateral connectivity within the retina, we may assume that the current adaptation level is a function of the pixel value itself as well as all other pixels in the image. This has given rise to all manner of spatially

FIGURE 8.3 *Simulation of chromatic adaptation in Reinhard and Devlin's photoreceptor-based operator. The level of chromatic adaptation may be approximated by setting user parameter c (shown here with values of 0.00, 0.50, 0.75, and 1.00).*

varying tone-reproduction models (see, for instance, Sections 8.1.2–8.1.4 and 8.2 for examples), but here a much faster and simpler solution is used, namely interpolation between pixel values and global averages:

$$I_{\mathrm{a}}(x, y) = a\, I_{\mathrm{r|g|b}}(x, y) + (1 - a)\, I^{\mathrm{av}}_{\mathrm{r|g|b}}$$

The interpolation weight a is user-specified and controls image appearance, which to some extent correlates with light adaptation, as shown in Figure 8.4. Plots of

FIGURE 8.4 Simulation of light adaptation in Reinhard and Devlin's operator. The level of light adaptation may be approximated by setting user parameter a to 0, 1/3, 2/3, and 1.

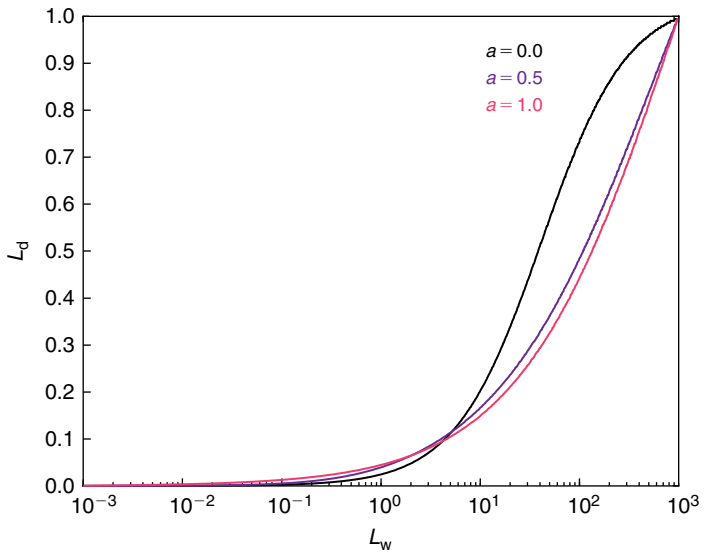


FIGURE 8.5 *Luminance mapping by the photoreceptor-based operator for different values of user parameter a.*

the operator for different values of a are presented in Figure 8.5. Both light and chromatic adaptation may be combined by bilinear interpolation

$$I_a^{\text{local}}(x, y) = c\, I_{r|g|b}(x, y) + (1 - c)\, L(x, y)$$

$$I_a^{\text{global}} = c\, I_{r|g|b}^{\text{av}} + (1 - c)\, L^{\text{av}}$$

$$I_a(x, y) = a\, I_a^{\text{local}}(x, y) + (1 - a)\, I_a^{\text{global}}$$

Directly inspired by photoreceptor physiology, this operator provides plausible results for a large class of images using default parameters. Most other results may

be optimized by adjusting the four user parameters. The value of c determines to what extent any color casts are removed, a and m affect the amount of contrast in the tone-mapped image, and f make the overall appearance lighter or darker. Because each of these parameters has an intuitive effect on the final result, manual adjustment is fast and straightforward.

8.1.2 PHOTOGRAPHIC TONE REPRODUCTION

The problem of mapping a range of world luminances to a smaller range of display luminances is not a new problem. Tone reproduction has existed in conventional photography since photography was invented. The goal of photographers is, often, to produce renderings of captured scenes that appear realistic. With photographic paper, like all paper, being inherently low dynamic range (LDR), photographers have to find ways to work around the limitations of the medium.

Although many common photographic principles were developed in the last 150 years and a host of media response characteristics were measured, a disconnect existed between the artistic and technical sides of photography. Ansel Adams's Zone System, which is still in use today, attempts to bridge this gap. It allows the photographer to use field measurements to improve the chances of creating a good final print.

The zone system may be used to make informed choices in the design of a tone-reproduction operator [274]. First, a linear scaling is applied to the image, which is analogous to setting exposure in a camera. Then, contrast may be locally adjusted using a computational model akin to photographic dodging-and-burning, which is a technique to selectively expose regions of a print for longer or shorter periods of time. This may bring up selected dark regions, or bring down selected light regions.

The key of a scene in photography is an indicator of how light or dark the overall impression of a scene is. Following other tone-reproduction operators, Reinhard et al. view the log average luminance $\bar{L}_w$ (Equation 7.7) as a useful approximation of a scene's key. For average-key scenes, the log average luminance should be mapped to 18% of the display range, which is in line with common photographic practice (although see the footnote on p. 64). Higher-key scenes should be mapped to a higher value and lower-key scenes should be mapped to a lower value. The value to

which the log average is mapped is given as a user parameter a. The initial scaling of the photographic tone-reproduction operator is then given by

$$L_{\mathrm{m}}(x, y) = \frac{a}{\bar{L}_{\mathrm{w}}} L_{\mathrm{w}}(x, y)$$

The subscript m denotes values obtained after the initial linear mapping. Because this scaling precedes any nonlinear compression, the operator does not necessarily expect the input to be specified in SI units. If the image is given in arbitrary units, the user parameter a could be adjusted accordingly. An example of this parameter's effect is shown in Figure 8.6. For applications that require hands-off operation, the value of this user parameter may be estimated from the histogram of the image [270]. This technique is detailed in Section 7.6.1.

Many scenes have a predominantly average dynamic range with a few high-luminance regions near highlights or in the sky. As a result of the chemistry involved, traditional photography uses s-shaped transfer functions (sigmoids) to compress both high and low luminance values while emphasizing the midrange. However, in modern photography, transfer functions are used that predominantly compress high luminances. This may be modeled with the following compressive function:

$$L_{\mathrm{d}}(x, y) = \frac{L_{\mathrm{m}}(x, y)}{1 + L_{\mathrm{m}}(x, y)}$$

This function scales small values linearly, whereas higher luminances are compressed by larger amounts. The function has an asymptote at 1, which means that all positive values will be mapped to a display range between 0 and 1. However, in practice, the input image does not contain infinitely large luminance values, and therefore the largest display luminances do not quite reach 1. In addition, it may be artistically desirable to let bright areas burn out in a controlled fashion. This effect may be achieved by blending the above transfer function with a linear mapping, yielding the following tone-reproduction operator:

$$L_{\mathrm{d}}(x, y) = \frac{L_{\mathrm{m}}(x, y)\left(1 + \frac{L_{\mathrm{m}}(x,y)}{L_{\mathrm{white}}^2}\right)}{1 + L_{\mathrm{m}}(x, y)}$$

This equation introduces a new user parameter, L_{white}, which denotes the smallest luminance value that will be mapped to white. By default, this parameter is set

FIGURE 8.6 *Prescaling the image data is an integral part of the photographic tone-reproduction operator, and it may be automated. Here, user parameter a was set to 0.01, 0.04, 0.18 (default), and 0.72.*

to the maximum world luminance (after the initial scaling). For lower dynamic range images, setting L_{max} to a smaller value yields a subtle contrast enhancement. Figure 8.7 shows various choices of L_{white} for an LDR image. Note that for hands-off operation, this parameter may also be estimated from the histogram of the input image [270].

The above equation is a reasonable global tone-reproduction operator. However, it may be modified to become a local tone-reproduction operator by applying an algorithm akin to photographic dodging-and-burning. For each pixel, we would

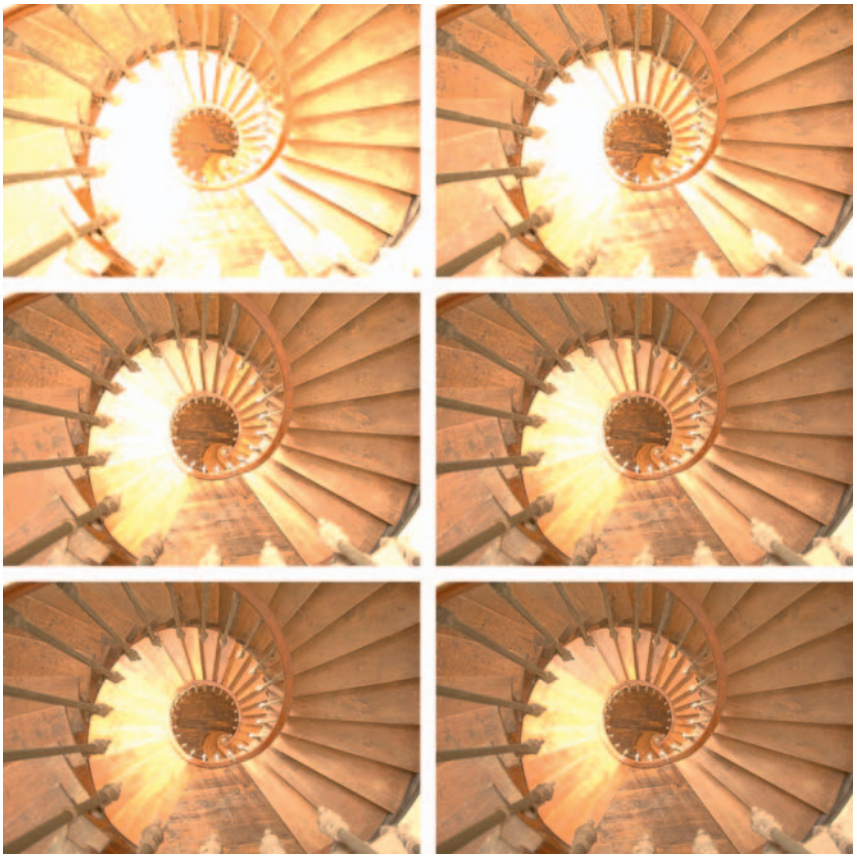


FIGURE 8.7 The L_{white} parameter in Reinhard et al.'s photographic tone-reproduction operator is effective in minimizing the loss of contrast when tone mapping an LDR image. The value of L_{white} was set to 0.15 in the top left image, and incremented by 0.10 for each subsequent image.

like to find the largest surrounding area that does not contain any sharp contrasts. A reasonable measure of contrast for this purpose is afforded by traditional center-surround computations. A Gaussian-weighted average is computed for a pixel (the center) and is compared with a Gaussian-weighted average over a larger region (the surround), with both centered over the same pixel. If there are no significant contrasts in the pixel's neighborhood, the difference of these two Gaussians will be close to 0. However, if there is a contrast edge that overlaps with the surround but not the center Gaussian, the two averages will be significantly different.

If a Gaussian-blurred image at scale s is given by

$$L_s^{\text{blur}}(x, y) = L_{\text{m}}(x, y) \otimes R_s(x, y),$$

the center-surround mechanism at that scale is computed with

$$V_s(x, y) = \frac{L_s^{\text{blur}} - L_{s+1}^{\text{blur}}}{2^{\Phi} a/s^2 + L_s^{\text{blur}}} \tag{8.1}$$

The normalization by $2^{\Phi} a/s^2 + L_s^{\text{blur}}$ allows this result to be thresholded by a common threshold that is shared by all scales because V_s is now independent of absolute luminance values. In addition, the $2^{\Phi} a/s^2$ term prevents the normalization from breaking for small values of L_s^{blur}. The user parameter Φ may be viewed as a sharpening parameter, and its effect is shown in Figure 8.8. For small values of Φ, its effect is very subtle. If the value is chosen too large, haloing artifacts may occur. In practice, a setting of $\Phi = 8$ yields plausible results.

This process yields a set of differences of Gaussians, each providing information about how much contrast is available within increasingly large areas around the pixel of interest. To find the largest area having relatively low contrast for a given pixel, we seek the largest scale s_{max} for which the difference of Gaussians (DoG) remains below a threshold:

$$s_{\text{max}} : |V_{s_{\text{max}}}(x, y)| < \varepsilon$$

For this scale, the corresponding center Gaussian may be taken as a local average. The local operator, which implements a computational model of dodging-and-burning,

FIGURE 8.8 *The sharpening parameter Φ in the photographic tone-mapping operator is chosen to be 4 and 8 (top row) and 16 and 32 (bottom row).*

is then given by

$$L_{\mathrm{d}}(x, y) = \frac{L_{\mathrm{m}}(x, y)}{1 + L_{s_{\max}}^{\mathrm{blur}}(x, y)}$$

The luminance of a dark pixel in a relatively bright region will satisfy $L < L_{s_{\max}}^{\mathrm{blur}}$, so this operator will decrease the display luminance L_{d}, thereby increasing the contrast at that pixel. This is akin to photographic "dodging." Similarly, a pixel in a relatively dark region will be compressed less, and is thus "burned." In either case, the pixel's contrast relative to the surrounding area is increased.

The memory efficiency of the dodging-and-burning version may be increased by realizing that the scale selection mechanism could be executed on the fly. The original implementation computes a Gaussian pyramid as a preprocess. Then, during tone mapping for each pixel, the most appropriate scale is chosen. Goodnight et al. show that the preprocessing step may be merged with the actual tone-reproduction stage, thus avoiding computing the low-pass images that will not be used [107]. Their work also shows how this operator may be implemented in graphics hardware.

In summary, the photographic tone-reproduction technique [274] exists in both global as well as local variants. For medium dynamic range images, the global operator is fast and provides sufficient compression. For very high dynamic range (HDR) images, local contrast may be preserved better with the local version, which implements dodging-and-burning. The local operator seeks for each pixel the largest area that does not contain significant contrast steps. This technique is, therefore, similar to edge-preserving smoothing filters such as the bilateral filter which is discussed in Section 8.5.2. We could, therefore, replace the scale selection mechanism with the more practical and efficient bilateral filter to produce a spatially localized average. This average would then serve the purpose of finding the average exposure level to which the pixel will be adjusted.

8.1.3 LOCAL SIGMOIDAL COMPRESSION

As argued in Section 7.4.2, the assumed adaptation level can be varied on a per-pixel basis. The photoreceptor-inspired model of Section 8.1.1 accomplishes this by linearly interpolating between an image average and the pixel value itself. Although

this is computationally very efficient, it is an approximation of what is likely to happen in the HVS. A closer approximation would be to compute local averages for each pixel to drive per-pixel semisaturation constants $\sigma(x, y)$. The Naka–Rushton equation (Equation 7.1) then takes on the general form:

$$\frac{R(x, y)}{R_{\max}} = \frac{I^n(x, y)}{I^n(x, y) + \sigma^n(x, y)}, \tag{8.2}$$

where $\sigma(x, y)$ is a function over a weighted average on a pixel neighborhood. This weighted average can be computed by applying a Gaussian filter [219]. The size of the filter determines whether halo artifacts occur. In general, smaller filter kernels provide fewer artifacts in this approach than larger kernels, as shown in Figure 8.9. Larger filter kernels also produce more contrast, as this figure shows. This is because of the fact that averaging over a small area around a given pixel produces an average that is likely to be similar to the pixel value itself. As a result, the value for σ and I

FIGURE 8.9 *The images were tone mapped with a spatially varying sigmoid, constructed by making the semisaturation constant dependent on a Gaussian-filtered image. The size of the filter kernel was increased in this sequence of images from 0.1 to 1, 10, and 100. The exponent n used in the sigmoidal compression was chosen to be 0.6. Note that the first image in the sequence appears most natural.*

will on average be relatively close, so that $R/R_{\max}$ will be mapped to a value close to 0.5. For larger filter kernels, this tends to happen less often in each image.

The filter artifacts can be reduced by using a filter that does not cross sharp boundaries. An edge-preserving smoothing operator, such as the bilateral filter, would serve this purpose, and can be combined with sigmoidal compression [181]. Here, $\sigma(x, y)$ is a function of the local adaptation value $I_b(x, y)$, which is computed using a bilateral-filtered image. The relationship between $I_b(x, y)$ and $\sigma(x, y)$ can be modeled with [373]:

$$\sigma(x, y) = I_b(x, y)^{\alpha} + \beta, \tag{8.3}$$

where I_b is computed using Equation 7.5. The constants α and β are empirical, and help adjust the semisaturation constant to control the overall appearance of the image. The value is 0.69 for α and β ranges from 2.0 to 5.83 [181].

The effect of adjusting the bilateral filter parameters is shown in Figures 8.10 and 8.11. The first of these shows two images created with identical parameters, with the exception of the kernel size of the intensity-domain Gaussian. It can be seen that the smaller of the two values produces an image with more detail, but slightly less global contrast.

FIGURE 8.10 *Sigmoidal compression using the bilateral filter to compute local adaptation values. The intensity-domain Gaussian used in the left image has a value of 1, whereas this value is 10 for the image on the right.*

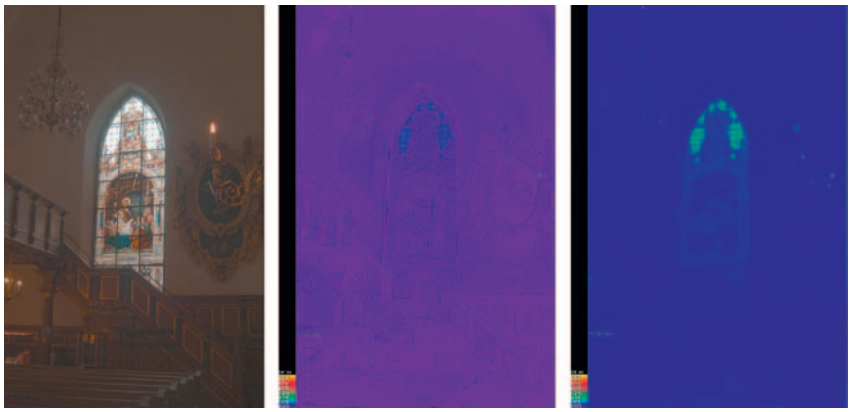


FIGURE 8.11 *The left image was computed using sigmoidal compression with local adaptation values provided by bilateral filtering. The spatial component of this filter is 1 pixel. The difference between this image and two others, computed with spatial kernels of 10 and 100 pixels, is shown in the middle and right respectively.*

The choice of spatial extent of the bilateral filter appears to be remarkably robust. We varied the size of the spatial Gaussian between 1 pixel and 100 pixels, keeping all other parameters constant. The results are shown in Figure 8.11. Indeed, as all three images would be very similar, we show only one image, with the remaining two panels showing the difference with this image, computed using the CIE dE94 color difference metric. The differences are well below the visible threshold, with the exception of the window panels.

8.1.4 ADAPTIVE GAIN CONTROL

Thus far, we have discussed several tone-reproduction operators that compute a local average. The photographic tone-reproduction operator uses a scale-space mechanism

to select how large a local area should be and computes a weighted average for this local area. It is then used to adjust the exposure level. Ashikhmin's operator does the same but provides an alternative explanation in terms of human vision. Similarly, the bilateral filter is effectively an edge-preserving smoothing operator. Smoothing by itself can be viewed as computing an average over a local neighborhood. The edge-preserving properties of the bilateral filter are important, because it allows the space over which the average is computed to be maximized.

The defining characteristic of the bilateral filter is that pixels are averaged over local neighborhoods, provided their intensities are similar. The bilateral filter is defined as

$$L^{\text{smooth}}(x, y) = \frac{1}{w(x, y)} \sum_u \sum_v b(x, y, u, v)\, L(x - u, y - v) \tag{8.4}$$

$$w(x, y) = \sum_u \sum_v b(x, y, u, v)$$

$$b(x, y, u, v) = f\left(\sqrt{(x - u)^2 + (y - v)^2}\right) g(L(x - u, y - v) - L(x, y))$$

with $w()$, a weight factor, normalizing the result and $b()$, the bilateral filter, consisting of components $f()$ and $g()$. There is freedom to choose the shape of the spatial filter kernel $f()$ as well as the luminance-domain filter kernel $g()$. Different solutions were independently developed in the form of nonlinear Gaussian filtering [14], SUSAN [299], and the bilateral filter [322]. At the same time, independent and concurrent developments led to alternative tone-reproduction operators—one based on the bilateral filter [74] (see Section 8.5.2) and the other based on the SUSAN filter [249].

Although Durand and Dorsey experimented with Gaussian filters as well as Tukey's filter, Pattanaik and Yee used a near box-shaped filter kernel in the luminance domain to steer the amount of compression in their tone-reproduction operator [249]. The latter used the output of their version of the bilateral filter as a local adapting luminance value, rather than as a mechanism to separate the image into a base and detail layer as Durand and Dorsey did.

Taking their cue from photography, Pattanaik and Yee note that white tends to be five times as intense as medium gray, and black is one-fifth the luminance of

medium gray. Their local gain control is derived from a weighted local average, where each surrounding pixel is weighted according to its luminance in relation to the luminance of the pixel of interest. Pixels more than five times as intense as the center pixel and pixels less than one-fifth its luminance are excluded from consideration.

For a circularly symmetric area around pixel (x, y), the local average is then computed for all pixels as follows:

$$\frac{1}{5} \leq \frac{L_{\mathrm{w}}(x-u, y-v)}{L_{\mathrm{w}}(x, y)} \leq 5$$

The circularly symmetric local area is determined by bounding the value of u and v by the radius r of the area under consideration:

$$\sqrt{u^2+v^2} \leq r$$

An alternative notation for the same luminance-domain constraint may be formulated in the log domain, with the base of the log being 5:

$$|\log_5(L_{\mathrm{w}}(x-u, y-v)) - \log_5(L_{\mathrm{w}}(x, y))| \leq 1$$

This implies a box filter in the luminance domain as well as a "box filter" (albeit circularly symmetric) in the spatial domain. A box filter in the luminance domain suffices if the image consists solely of sharp edges. Smoother high-contrast edges are best filtered with a luminance-domain filter, which has a somewhat less abrupt cutoff. This may be achieved with the following luminance-domain filter kernel $g()$:

$$g(x-u, y-v) = \exp\left(-|\log_5(L_{\mathrm{w}}(x-u, y-v)) - \log_5(L_{\mathrm{w}}(x, y))|^{25}\right)$$

The spatial filter kernel $f()$ is circularly symmetric and unweighted:

$$f(x-u, y-v) = \begin{cases} 1 & \text{if } \sqrt{u^2+v^2} \leq r \\ 0 & \text{otherwise} \end{cases}$$

The result of producing a filtered image with the above filter is an image that is blurred, except in areas where large contrast steps occur. This filter may, therefore,

be viewed as an edge-preserving smoothing filter, just as the bilateral and trilateral filters. The output of this filter may therefore be used in a manner similar to tone-reproduction operators that split an image into a base layer and a detail layer. The base layer is then compressed and recombined with the detail layer under the assumption that the base layer is HDR and the detail layer is LDR.

Alternatively, the output of this filter may be viewed as a local adapting luminance. Any of the global operators that make use of a global average may thus be extended to become local operators. The output of any edge-preserving smoothing operator as well as the scale selection mechanism of Reinhard et al.'s photographic operator may replace the global average with a local average. In each case, the typical trade-off between amount of achievable compression and visibility of haloing artifacts will return. However, by using edge-preserving smoothing operators or the aforementioned scale selection mechanism, the local average is ensured to be relatively close to the pixel value itself. Although halos may not be avoided altogether, they are minimized with these approaches.

8.2 IMAGE APPEARANCE MODELS

Although the color appearance models have evolved to predict the perception of color under illumination that may vary in both strength and chromatic content, as a result of the spatial nature of human visual processing, they are not suitable for direct application to images.[1] Extending color appearance models for the purpose of predicting how an image would be perceived under a given illumination is currently an active area of research. In this section, we discuss several such models in the order in which they were developed.

8.2.1 MULTISCALE OBSERVER MODEL

Pattanaik's multiscale observer model ranks among the more complete color appearance models and consists of several steps that are executed in succession [246].

1 For instance, a patch of color is assumed to be surrounded by a background and a surround. In images, a pixel is surrounded by other pixels which can have any values. Color appearance models are not designed to deal with such complex environments.

The output of this model (and all other color appearance models) are color appearance correlates, as discussed in Section 2.9. A tone-reproduction operator may be derived from these correlates by executing the inverse model and substituting characteristics of the display device into the equations in the appropriate place.

For simplicity, we present a version of the model which is reduced in complexity. For the purpose of tone reproduction, some of the forward and backward steps of the model cancel out and may, therefore, be omitted. Also, compared with the original model, we make small changes to minimize visual artifacts, for instance, by choosing the filter kernel sizes smaller than in the original model. We first give a brief overview of the full model and then detail a simplified version.

The first step in the forward model is to account for light scatter in the ocular media, followed by spectral sampling to model the photoreceptor output. This yields four images representing the rods and the L, M, and S cones. These four images are then each spatially decomposed into seven-level Gaussian pyramids and subsequently converted into four six-level DoG stacks that represent band-pass behavior as seen in the HVS. DoGs are computed by subtracting adjacent images in the pyramid.

The next step consists of a gain-control system applied to each of the DoGs in each of the four channels. The shape of the gain-control function resembles threshold-versus-intensity curves such that the results of this step may be viewed as adapted contrast pyramidal images. The cone signals are then converted into a color opponent scheme that contains separate luminance, red–green, and yellow–blue color channels. The rod image is retained separately.

Contrast transducer functions that model human contrast sensitivity are then applied. The rod and cone signals are recombined into an achromatic channel as well as red–green and yellow–blue color channels. A color appearance map is formed next, which is the basis for the computation of the aforementioned appearance correlates. This step cancels in the inverse model, and we therefore omit a detailed description. We also omit computing the rod-signals since we are predominantly interested in photopic lighting conditions.

The model calls for low-pass filtered copies with spatial frequencies of 0.5, 1, 2, 4, 8, and 16 cycles per degree (cpd). Specifying spatial frequencies in this manner is common practice when modeling the HVS. However, for a practical tone-reproduction operator, this would require knowledge of the distance of the observer

to the display device as well as the spatial resolution of the display device. Because viewer distance is difficult to control, let alone anticipate, we restate spatial frequencies in terms of cycles per pixel.

Furthermore, we omit the initial modeling of light scatter in the ocular media. Modeling light scatter would have the effect of introducing a small amount of blur to the image, particularly near areas of high luminance. On occasions, modeling of glare may be important and desirable, and should be included in a complete implementation of the multiscale observer model. However, for simplicity, we omit this initial processing. This set of simplifications allows us to focus on the part of the multiscale observer model that achieves dynamic range reduction.

The model expects input to be specified in LMS cone space, which is discussed in Section 2.4. The compressive function applied in all stages of the multiscale observer model is given by the following gain control:

$$G(L) = \frac{1}{0.555(L+1)^{0.85}}$$

Multiplying either a low-pass or band-pass image by this gain control amounts to applying a sigmoid.

A stack of seven increasingly blurred images is created next. The amount of blur is doubled at each level, and for the smallest scale we use a filter kernel the size of which is determined by a user parameter which is discussed later in this section. An image at level s is represented with the following triplet:

$$\left(L_s^{\text{blur}}(x,y),\ M_s^{\text{blur}}(x,y),\ S_s^{\text{blur}}(x,y)\right)$$

From this stack of seven Gaussian-blurred images we may compute a stack of six DoGs images, which represent adapted contrast at six spatial scales:

$$L_s^{\text{DoG}}(x,y) = \left(L_s^{\text{blur}}(x,y) - L_{s+1}^{\text{blur}}(x,y)\right)\ G\left(L_{s+1}^{\text{blur}}(x,y)\right)$$

$$M_s^{\text{DoG}}(x,y) = \left(M_s^{\text{blur}}(x,y) - M_{s+1}^{\text{blur}}(x,y)\right)\ G\left(M_{s+1}^{\text{blur}}(x,y)\right)$$

$$S_s^{\text{DoG}}(x,y) = \left(S_s^{\text{blur}}(x,y) - S_{s+1}^{\text{blur}}(x,y)\right)\ G\left(S_{s+1}^{\text{blur}}(x,y)\right)$$

The DoGs scheme involves a division by a low-pass filtered image (through the gain-control function), which may be viewed as a normalization step. This approach was followed in both Ashikhmin's operator (see Section 8.3.3) as well as the photographic tone-reproduction operator (Section 8.1.2). DoGs are reasonable approximations of some of the receptive fields found in the HVS.[2] They are also known as "center-surround mechanisms."

The low-pass image at level $s = 7$ is retained and will form the basis for image reconstruction. In the final step of the forward model, pixels in this low-pass image are adapted to a linear combination of themselves and the mean value ($\bar{L}_7^{\text{blur}}$, $\bar{M}_7^{\text{blur}}$, $\bar{S}_7^{\text{blur}}$) of the low-pass image:

$$L_7^{\text{blur}}(x, y) = L_7^{\text{blur}}(x, y)\ G\Big((1 - A)\,\bar{L}_7^{\text{blur}} + A\,L_7^{\text{blur}}(x, y)\Big)$$

$$M_7^{\text{blur}}(x, y) = M_7^{\text{blur}}(x, y)\ G\Big((1 - A)\,\bar{M}_7^{\text{blur}} + A\,M_7^{\text{blur}}(x, y)\Big)$$

$$S_7^{\text{blur}}(x, y) = S_7^{\text{blur}}(x, y)\ G\Big((1 - A)\,\bar{S}_7^{\text{blur}} + A\,S_7^{\text{blur}}(x, y)\Big)$$

The amount of dynamic range reduction is determined by user parameter A in the above equations, which takes a value between 0 and 1. The effect of this parameter on the appearance of tone-mapped images is shown in Figure 8.12.

The forward version of the multiscale observer model is based on the HVS. While we could display the result of the forward model, the viewer's visual system would then also apply a similar forward model (to the extent that this model is a correct reflection of the HVS). To avoid applying the model twice, the computational model should be inverted before an image can be displayed. During the reversal process, parameters pertaining to the display device are inserted in the model so that the result is ready for display.

In the first step of the inverse model, the mean luminance $L_{\text{d,mean}}$ of the target display device needs to be determined. For a typical display device, this value may

2 A receptive field may be seen as the pattern of light that needs to be present to optimally stimulate a cell in the visual pathway.

FIGURE 8.12 *Using the multiscale observer model, the interpolation parameter A was set to 0, 0.25, 0.50, 0.75, and 1.*

be set to around 50 cd/m^2. A gain-control factor for the mean display luminance is determined, and the low-pass image is adapted once more, but now for the mean display luminance:

$$L_7^{\text{blur}}(x, y) = \frac{L_7^{\text{blur}}(x, y)}{G(L_{\text{d,mean}})}$$

$$M_7^{\text{blur}}(x, y) = \frac{M_7^{\text{blur}}(x, y)}{G(M_{\text{d,mean}})}$$

$$S_7^{\text{blur}}(x, y) = \frac{S_7^{\text{blur}}(x, y)}{G(S_{\text{d,mean}})}$$

The stack of DoGs is then added to the adapted low-pass image one scale at a time, starting with $s = 6$ followed by $s = 5, 4, \ldots, 0$:

$$L_7^{\text{blur}}(x, y) = \max\left(L_7^{\text{blur}}(x, y) + \frac{L_s^{\text{DoG}}(x, y)}{G(L_7^{\text{blur}}(x, y))},\ 0\right)$$

$$M_7^{\text{blur}}(x, y) = \max\left(M_7^{\text{blur}}(x, y) + \frac{M_s^{\text{DoG}}(x, y)}{G(M_7^{\text{blur}}(x, y))},\ 0\right)$$

$$S_7^{\text{blur}}(x, y) = \max\left(S_7^{\text{blur}}(x, y) + \frac{S_s^{\text{DoG}}(x, y)}{G(S_7^{\text{blur}}(x, y))},\ 0\right)$$

Finally, the result is converted to XYZ and then to RGB, where gamma correction is applied.

The original formulation of this model shows haloing artifacts similar to those created by other local operators discussed in this chapter. One of the reasons for this is that the model is calibrated in degrees of visual angle rather than in pixels. The transformation between degrees of visual angle to pixels requires assumptions on the size of the display, its resolution, and the distance between the observer and the display. The size of the filter kernel used to create the low-pass images is directly affected by these assumptions. For the purpose of demonstration, we show a sequence of images produced with different kernel sizes in Figure 8.13. Note that we only adjust the size of the smallest Gaussian. By specifying the kernel size for the smallest Gaussian, the size of all other Gaussians is determined. The figure shows that smaller Gaussians produce smaller halos, which are less obtrusive than the larger halos of the original model.

The reconstruction of a displayable image proceeds by successively adding band-pass images back to the low-pass image. These band-pass images by default receive equal weight. It may be beneficial to weight band-pass images such that higher spatial frequencies contribute more to the final result. Although the original multi-scale observer model does not feature such a weighting scheme, we have found that

FIGURE 8.13 *Using the multiscale observer model, the filter kernel size is set to 0.06, 0.12, 0.25, and 0.5 in this sequence of images.*

contrast in the final result may be improved if higher frequencies are given a larger weight. This is shown in Figure 8.14, where each successive image places more emphasis on higher frequencies. The scale factor k used for these images relates to the index number s of the band-pass pyramid in the following manner:

$$k = (6 - s)\, g$$

FIGURE 8.14 *Relative scaling in the multiscale observer model. For a filter kernel size of 0.03, the relative scaling parameter was set to 2, 3, 4, and 5.*

The constant g is a user parameter, which we vary between 1 and 5 in Figure 8.14. A larger value for g produces more contrast in the tone-mapped image, but if this value is set too high, the residual halos present in the image are emphasized also, which is generally undesirable.

For uncalibrated images tone mapped with the multiscale observer model, different prescale factors cause the overall image appearance to be lighter or darker, as shown in Figure 8.15.

The computational complexity of this operator remains high, and we would only recommend this model for images with a very HDR. If the amount of compression required for a particular image is less, simpler models likely suffice. The Fourier transforms used to compute the low-pass images are the main factor that determine running time. There are seven levels in the Gaussian pyramid and four color channels

FIGURE 8.15 *Effect of prescaling on the multiscale observer model. Images are prescaled by factors of 0.01, 0.1, 1, and 10.*

in the original model, resulting in 28 low-pass filtered images. In our simplified model, we only compute three color channels, resulting in a total of 21 low-pass images.

The multiscale observer model is the first to introduce center-surround processing to the field of tone reproduction, which is also successfully used in Ashikhmin's operator (see following section) as well as Reinhard et al.'s photographic tone-reproduction operator (see Section 8.1.2). The halos present in the original model may be minimized by carefully choosing an appropriate filter kernel size.

8.2.2 iCAM

Although most operators in this chapter are aimed at dynamic range reduction, Pattanaik's multiscale observer model [246] (discussed in the following section) and Fairchild's iCAM model [84] are both color appearance models.

Most color appearance models, such as CIECAM97, CIECAM02, and the Hunt model, are intended for use in simplified environments. It is normally assumed that a uniform patch of color is viewed on a larger uniform background with a different color. The perception of this patch of color may then be predicted by these models with the XYZ tristimulus values of the patch and a characterization of its surround as input, as described in Section 2.9.

Images tend to be more complex than just a patch on a uniform background. The interplay between neighboring pixels may require a more complex spatially variant model, which accounts for the local adaptation of regions around each pixel. This argument in favor of spatially variant color appearance models is effectively the same as the reasoning behind spatially variant tone-reproduction operators. The parallels between the iCAM model described here and, for instance, Chiu's and Rahman's operators are therefore unmistakable. However, there are also sufficient differences to make a description of the model worthwhile.

The iCAM "image appearance model" is a direct refinement and simplification of the CIECAM02 color appearance model [84,143]. It omits the sigmoidal compression found in CIECAM02 but adds spatially variant processing in the form of two separate Gaussian-blurred images that may be viewed as adaptation levels. Like most color appearance models, the model needs to be applied both in forward direction and inverse direction.

The input to the model is expected to be specified in XYZ device-independent coordinates. Like CIECAM02, the model uses various color spaces to execute the different stages of the algorithm. The first stage is a chromatic adaptation transform, for which sharpened cone responses are used. Sharpened cone responses are obtained with the M_{CAT02} transform, given in Section 2.4.

The chromatic adaptation transform pushes the colors in the image toward the D_{65} white point. The amount of adaption in this von Kries transform is determined by a user parameter D, which specifies the degree of adaptation. In addition, for each pixel, a white point $W(x, y)$ is derived from the XYZ image by applying a low-pass filter with a kernel a quarter the size of the image. This may be applied to each color channel independently for chromatic adaptation, or on the Y channel only for achromatic adaptation. This low-pass filtered image is then also converted with the M_{CAT02} matrix. Finally, the D_{65} white point, given by the $Y_{\text{w}} = 95.05, 100.0, 108.88$ triplet, is also converted to sharpened cone responses. The subsequent von Kries adaptation transform is given by

$$R_{\text{c}}(x, y) = R'(x, y)\left(Y_{\text{w}}\frac{D}{W_{R'}(x, y)} + 1 - D\right)$$

$$G_{\text{c}}(x, y) = G'(x, y)\left(Y_{\text{w}}\frac{D}{W_{G'}(x, y)} + 1 - D\right)$$

$$B_{\text{c}}(x, y) = B'(x, y)\left(Y_{\text{w}}\frac{D}{W_{B'}(x, y)} + 1 - D\right)$$

This transform effectively divides the image by a filtered version of the image. This step of the iCAM model is, therefore, similar to Chiu's and Rahman's operators. In those operators, the trade-off between amount of available compression and presence of halos was controlled by a scaling factor k. Here, D plays the role of the scaling factor. We may, therefore, expect this parameter to have the same effect as k in Chiu's and Rahman's operators. However, in the above equation, D also determines the amount of chromatic adaptation. It serves the same role as the degree of adaptation parameter found in other color appearance models (compare, for instance, with CIECAM02, which is described in Section 2.9).

For larger values of D, the color of each pixel is pushed closer to the D_{65} white point. Hence, in the iCAM model, the separate issues of chromatic adaptation, haloing, and amount of compression are directly interrelated.

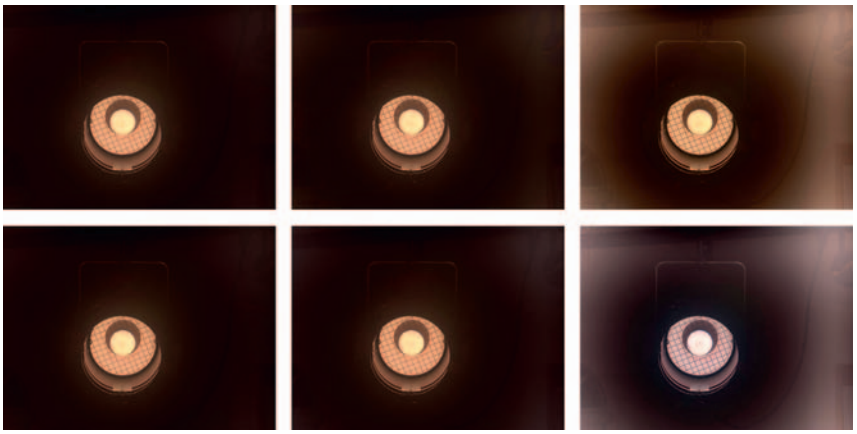


FIGURE 8.16 *The iCAM image appearance model. Top row: luminance channel used as adaptation level for all three channels. Bottom row: channels are processed independently. From left to right, the adaptation parameter D was varied from 0.0 to 0.5 and 1.0.*

Figure 8.16 shows the effect of parameter D, which was given values of 0.0, 0.5, and 1.0. This figure also shows the effect of computing a single white point, which is shared among the three values of each pixel, and computing a separate white point for each color channel independently. For demonstration purposes, we have chosen an image with a higher dynamic range than usual. The halo visible around the light source is therefore more pronounced than for images with a medium dynamic range. Like Chiu's and Rahman's operators, the iCAM model appears most suited for medium dynamic range images.

After the chromatic adaptation transform, further compression is achieved by an exponential function, which is executed in LMS cone space (see Section 2.4). The

exponential function, which compresses the range of luminances, is given by

$$L'(x, y) = |L(x, y)|^{0.43\, F_{\mathrm{L}}(x,y)}$$
$$M'(x, y) = |M(x, y)|^{0.43\, F_{\mathrm{L}}(x,y)}$$
$$S'(x, y) = |S(x, y)|^{0.43\, F_{\mathrm{L}}(x,y)}$$

The exponent is modified on a per-pixel basis by F_{L}, which is a function of a spatially varying surround map that is derived from the luminance channel (Y channel) of the input image. The surround map $S(x, y)$ is a low-pass filtered version of this channel with a Gaussian filter kernel size of one-third the size of the image. The function F_{L} is then given by

$$F_{\mathrm{L}}(x, y) = \frac{1}{1.7}\left(0.2\left(\frac{1}{5\, S(x, y) + 1}\right)^4 (5\, S(x, y)) + 0.1\left(1 - \left(\frac{1}{5\, S(x, y)}\right)^4\right)^2 \sqrt[3]{5\, S(x, y)}\right)$$

Thus, this computation of F_{L} may be seen as the spatially variant extension of CIECAM02's factor for partial adaptation, given in Equation 2.2.

This step completes the forward application of the iCAM model. To prepare the result for display, the inverse model should be applied. The model requires the same color spaces to be used as in the forward model in each of the steps. The first step is to invert the above exponentiation:

$$L'(x, y) = |L(x, y)|^{1/0.43}$$
$$M'(x, y) = |M(x, y)|^{1/0.43}$$
$$S'(x, y) = |S(x, y)|^{1/0.43}$$

The inverse chromatic adaptation transform does not require a spatially variant white point, but converts from a global D_{65} white point $Y_{\mathrm{w}} = 95.05, 100.0, 108.88$ to an

equiluminant white point $Y_e = 100, 100, 100$. Because full adaptation is assumed, D is set to 1 and this transform simplifies to the following scaling, which is applied in sharpened cone response space:

$$R' = R \frac{Y_e}{Y_w}$$

$$G' = G \frac{Y_e}{Y_w}$$

$$B' = B \frac{Y_e}{Y_w}$$

After these two steps are executed in their appropriate color spaces, the final steps consist of clipping the 99th of all pixels, normalization, and gamma correction.

The user parameters for this model are D, as discussed earlier, and a prescaling of the input image. This prescaling may be necessary because the iCAM model requires the input to be specified in candelas per square meter. For arbitrary images, this requires the user to scale the image to its appropriate range before tone mapping. The effect of prescaling is shown in Figure 8.17. For images that contain values that are too small, a red shift is apparent. If the values in the image are too large, the overall appearance of the image becomes too dark.[3]

Further parameters for consideration are the kernel sizes of the two Gaussian filters. For the images shown in this section, we used the recommended kernel sizes of $\frac{1}{4}$ and $\frac{1}{3}$ the size of the image, but other sizes are possible. As with Chiu's and Rahman's operators, the precise kernel size is unimportant, as long as the filter width is chosen to be large.

FIGURE 8.17 *Effect of prescaling on the iCAM model. The luminance range for these images was* 10^3, 10^4, 10^5, *and* 10^6 cd/m^2.

3 The images in this figure, as with similar image sequences for other operators, were scaled beyond a reasonable range—both too small and too large—to show the effect of the parameter. It should be noted that, in practice, a reasonable parameter setting should be chosen to avoid such extremes.

In summary, the iCAM model consists of two steps: a chromatic adaptation step followed by an exponential function. The chromatic adaptation step strongly resembles Chiu's and Rahman's operators since the image is divided by a blurred version of the image. The second step may be viewed as an advanced form of gamma correction, whereby the gamma factor is modulated on a per-pixel basis. The forward model needs to be followed by the inverse application of the model to prepare the image for display. A final clipping and normalization step brightens the overall appearance. The model is best suited for images with a medium dynamic range because the trade-off between compression and presence of halos is less critical for this class of images than for very HDR images.

8.2.3 iCAM06

Tone-reproduction evaluations have shown that the iCAM image appearance model described earlier does not perform as well as some other operators [168]. In particular, it was found that less local contrast and colorfulness was present in the tone-mapped results as compared with the input images. Consequently, the authors have updated their model and designated it iCAM06 [168].

This new model retains the local white-balancing operator, as well as chromatic adaptation, and the use of the IPT uniform color space. However, the single-scale Gaussian filtering was replaced by using the bilateral filter, and therefore incorporates a dual-layer approach (see Section 2.7.3). The nonlinear gamma correction step was replaced with a more common photoreceptor-response function. It also adds scotopic as well as photopic processing. Finally, it adds modules for modeling the Stevens, Hunt, and Bartleson–Breneman effects. The flowchart for iCAM06 is given in Figure 8.18.

The model assumes that the input is specified in XYZ space. Then, the image is decomposed into a base XYZ_{base} and detail layer XYZ_{detail} by means of bilateral filtering [322]. The base layer will have an HDR, whereas the detail layer tends to have LDR but high-frequency details. The detail layer to some extent corresponds to reflectances found in a scene, whereas it is surmised that the base layer roughly corresponds to illumination [168]. Given that the human vision is mostly sensitive to reflectance, it is the base layer that is subsequently compressed, while being able to maintain the perceptual appearance of the final result.

Dynamic range reduction by splitting the image into a base and detail layer was first proposed by Durand and Dorsey [74], as discussed in Section 8.5.2. However, a

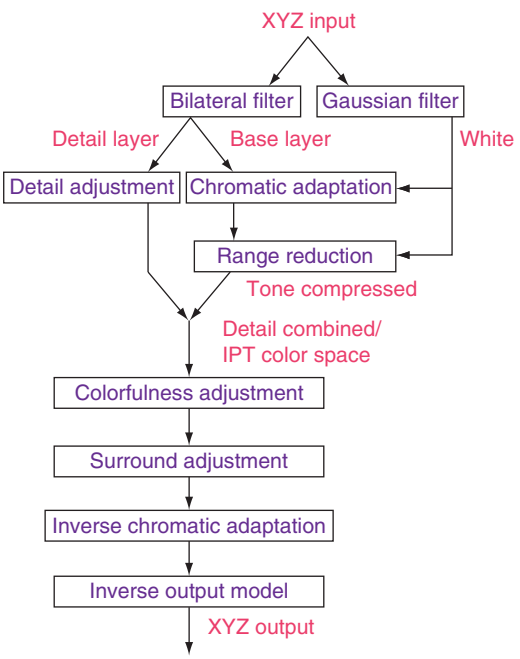


FIGURE 8.18 *Flowchart outlining the steps comprising the iCAM06 model.*

crucial difference is that dynamic range reduction is achieved by sigmoidal compression rather than by linear scaling. Moreover, the compression step is preceded by chromatic adaptation. For photopic signals, these two steps are effectively the same as the first two steps found in CIECAM02 and use the same set of color spaces (see Section 2.9), albeit that the exponent of 0.42 in the nonlinear compression step is replaced with a user-controllable parameter p, leading to compression of the form:

$$R'_a = \frac{400\left(F_L\ R'_{base}/100\right)^p}{27.13 + \left(F_L\ R'_{base}/100\right)^p} + 0.1$$

$$G'_{\mathrm{a}} = \frac{400\left(F_{\mathrm{L}}\, G'_{\mathrm{base}}/100\right)^{p}}{27.13 + \left(F_{\mathrm{L}}\, G'_{\mathrm{base}}/100\right)^{p}} + 0.1$$

$$B'_{\mathrm{a}} = \frac{400\left(F_{\mathrm{L}}\, B'_{\mathrm{base}}/100\right)^{p}}{27.13 + \left(F_{\mathrm{L}}\, B'_{\mathrm{base}}/100\right)^{p}} + 0.1$$

The value of the exponent p, which controls the steepness of the tone compression curve, is typically in the range 0.6–0.85. Its effect is shown in a sequence of images in Figure 8.19.

Under scotopic vision, only the rods are active, and iCAM06 models this behavior as follows. After chromatic adaptation, the luminance L_S is computed for each pixel. A reference white is also passed through the chromatic adaptation module, leading to a reference luminance L_{S_W}. The rod response A_S is then computed with the following set of equations:

$$L_{\mathrm{LS}} = 5L_A$$

$$j = \frac{10^{-4}}{L_{\mathrm{LS}} + 10^{-4}}$$

$$B_S = \frac{0.5}{1 + 0.3\left(L_{\mathrm{LS}}\dfrac{L_S}{L_{S_W}}\right)^{0.3}} + \frac{0.5}{1 + 5\,L_{\mathrm{LS}}}$$

$$F_{\mathrm{LS}} = 3800\, j^2 L_{\mathrm{LS}} + 0.2\left(1 - j^2\right)^4 L_{\mathrm{LS}}^{1/6}$$

$$A_S = 3.05\, B_S \frac{400\left(F_{\mathrm{LS}}\dfrac{L_S}{L_{S_W}}\right)^{p}}{27.13 + \left(F_{\mathrm{LS}}\dfrac{L_S}{L_{S_W}}\right)^{p}} + 0.3$$

FIGURE 8.19 *The steepness of the tone compression curve, controlled by user parameter p, affects the overall amount of contrast. In the images on the next page, the value of p was set to 0.4, 0.6, 0.8, and 1.0 (the recommended range for this parameter is [0.6, 0.85].)*

Here, L_{LS} is a quantity related to the scotopic luminance, B_S is the rod pigment bleach or saturation factor, and F_{LS} is the scotopic luminance level adaptation factor. These computations stem from Hunt's color appearance model [134]. The rod response A_S is then added to the tone-mapped cone responses:

$$R_{\mathrm{TC}} = R'_{\mathrm{a}} + A_S$$
$$G_{\mathrm{TC}} = G'_{\mathrm{a}} + A_S$$
$$B_{\mathrm{TC}} = B'_{\mathrm{a}} + A_S$$

These values are subsequently converted back to XYZ space, and recombined with the detail layer, which by itself has been adjusted to model the Stevens effect:

$$f = (F_L + 0.8)^{0.25}$$
$$X'_{\mathrm{detail}} = X^f_{\mathrm{detail}}$$
$$Y'_{\mathrm{detail}} = Y^f_{\mathrm{detail}}$$
$$Z'_{\mathrm{detail}} = Z^f_{\mathrm{detail}}$$

The luminance-dependent exponent models the Stevens effect, which predicts that brightness contrast or lightness contrast (i.e., perceived quantities) increase with an increase in luminance.

The recombined signal is then subjected to the IPT color space transform (see Section 2.7.3), for the purpose of adjusting image attributes that improve the prediction of appearance effects. In particular, the P and T channels are modified to predict the Hunt effect (perceived colorfulness increases with luminance). To this end, chroma C is computed first:

$$C = \sqrt{P^2 + T^2}$$

followed by a chroma- and luminance-dependent scalar g:

$$g = (F_{\mathrm{L}} + 1)^{0.2} \frac{1.29\,C^2 - 0.27\,C + 0.42}{C^2 - 0.31\,C + 0.42}$$

which is used to scale the P and T channels:

$$P' = Pg$$
$$T' = Tg$$

Finally, the I channel is adjusted to compensate for the fact that in different viewing conditions (dark, dim, and average conditions are distinguished, as in CIECAM02), the perceived image contrast changes with viewing condition:

$$I' = \begin{cases} I^{1.5} & \text{Dark surround} \\ I^{1.25} & \text{Dim surround} \\ I^{1.0} & \text{Average surround} \end{cases}$$

To prepare this result for display, the IPT image is converted to XYZ, an inverse chromatic adaptation is applied to take the data from D65 to the white point of the display, and the image is then transformed to the output device's RGB color space. The final rendering is aided by clipping the 1st and 99th percentile of the image data.

8.2.4 COLOR APPEARANCE UNDER EXTENDED LUMINANCE LEVELS

Color appearance models are validated by matching against psychophysical data. In particular, the LUTCHI data set [198,199] has been used to optimize CIECAM97 and CIECAM02. This data set consists of a large set of stimuli (XYZ tristimulus values with environment descriptions), with corresponding appearance correlates, which were obtained by means of a magnitude estimation experiment. Although this experiment was repeated for different media and for different environment conditions, in particular, the strength of illumination was relatively limited.

Recently, a similar experiment was performed, except that high luminance levels up to $16,860\,\text{cd/m}^2$ were included [156]. Such an experiment can be conducted using an HDR display device. The particular device used here consists of a light box with two 400-W Hydrargyrum Medium-Arc Iodide (HMI) bulbs, which transmit light through a set of filters, which then illuminates either a 19-inch liquid crystal display (LCD) panel or a diffuser with transparencies. The HMI bulbs emit light at a correlated color temperature of approximately 6500 K, and stay cool enough to

prevent the device from overheating. This assembly is capable of delivering up to $30{,}000\,\mathrm{cd/m^2}$. By including combinations of neutral density filters, the device can be tuned to emit peak luminance values of 50, 125, 500, 1000, 2200, 8500, and $16{,}860\,\mathrm{cd/m^2}$. Note that this setup does not provide a high number of luminance levels; the device is still 8-bit, but for the purpose of the magnitude-estimation experiment, this is sufficient dynamic range.

The magnitude-estimation experiment closely follows the paradigm used for the LUTCHI data set. A viewing pattern as shown in Figure 8.20 is presented to trained expert observers. Along the border are decorating patches, which serve to create a stimulus with a reasonable distribution of colors. The three patches in the middle are used for the experiment. The bottom patches are shown to provide the observer with a reference lightness and colorfulness. For the top patch, the observer estimates lightness, colorfulness, and hue on appropriately chosen scales.

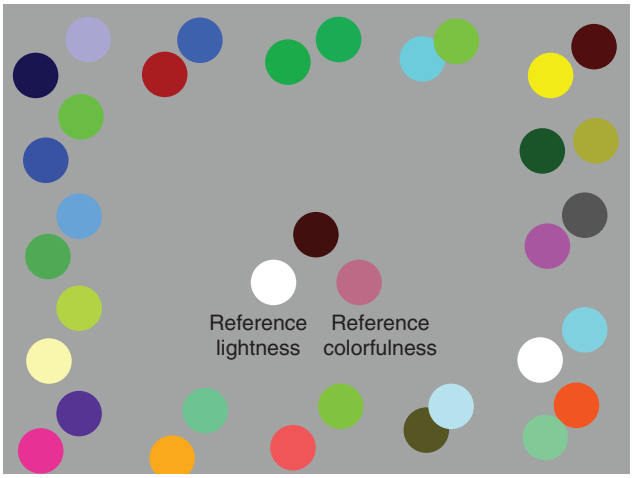


FIGURE 8.20 *An example of the viewing pattern observed during the magnitude-estimation experiment (after [156]).*

By repeating this experiment for a sufficient number of randomly chosen patches under a range of viewing conditions, a data set can be constructed. For each patch, this data set contains an XYZ tristimulus value describing the colorimetric value of the patch, as well as a description of the viewing environment on the one hand, and an estimate of the associated appearance correlates of lightness, hue, and colorfulness, averaged over all participants in the experiment.

Some of the results of this experiment are reproduced in Figure 8.21. The left panel shows that at higher peak luminance levels, both the correlates of lightness and colorfulness increase. The latter is known as the "Hunt effect." The panel on the right shows that if the luminance level is kept constant, but the background level is increased, the measured lightness and colorfulness decrease.

These findings can be used to guide the design of a color appearance model, valid under high luminance levels [156]. This model aims to predict lightness, colorfulness, and hue, as well as the Hunt effect, the Stevens effect (lightness contrast

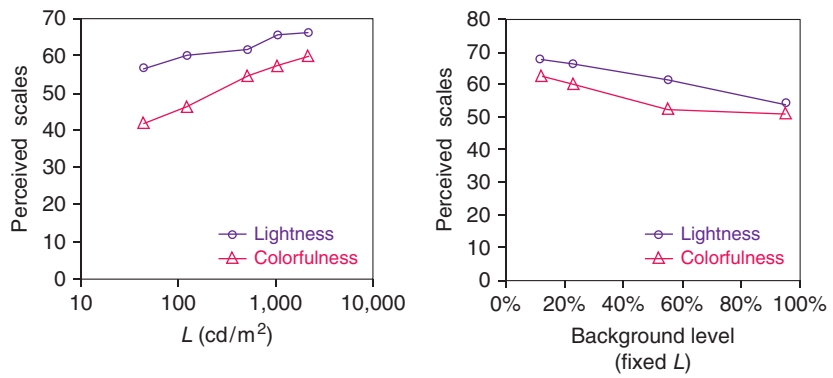


FIGURE 8.21 *Results of the magnitude-estimation experiment. On the left, the luminance was varied while keeping the background fixed, whereas on the right, the luminance was fixed while varying the background (after [156]).*

increases with luminance level), and simultaneous contrast (lightness and colorfulness change with background luminance levels).

The model consists of four components, which are chromatic adaptation and cone response, followed by the computation of achromatic and chromatic responses. The chromatic adaptation transform is identical to the CAT02 transform, which is discussed in Section 2.5. The nonlinear response compression is mediated by the aforementioned Naka–Rushton equation (Equation 7.1), executed independently on the three color channels of the Hunt–Pointer–Estevez color space (which is effectively cone response space, defined in Section 2.9.1). The semisaturation constant σ in 7.1 is replaced with the absolute level of adaptation L_a, and the exponent n is chosen to be 0.57:

$$L' = \frac{L^{0.57}}{L^{0.57} + L_a^{0.57}}$$

$$M' = \frac{M^{0.57}}{M^{0.57} + M_a^{0.57}}$$

$$S' = \frac{S^{0.57}}{S^{0.57} + S_a^{0.57}}$$

For the purpose of computing an achromatic response A, the different cone responses are combined, assuming a ratio of 40:20:1 between cone types:

$$A = \frac{40L' + 20M' + S'}{61}$$

The same achromatic response can be computed for the reference white, yielding A_W. The ratio between A and A_W is related to lightness J', which Kim et al. determined to be as follows:

$$J' = \left(\frac{-\left(-\frac{A}{A_W} - 0.24\right) 0.208}{\frac{A}{A_W} - 1.13} \right)^{0.274}$$

where the constants arise from fitting the model to the experimental data described earlier in this section. Negative values of J' should be clamped. As lightness depends

on the type of output medium, this measure can be improved by applying the following correction:

$$J = 100\big(E\left(J' - 1\right) + 1\big)$$

The correction factor E is a constant that depends on the type of output medium: $E = 1.0$ for high-luminance LCD displays, $E = 1.2175$ for transparent output media, $E = 1.4572$ for cathode ray tubes (CRT) displays, and for reflective paper, $E = 1.7526$.

Following other color appearance models, chroma can be computed from a color opponent pair (a, b), as follows:

$$a = \frac{11L' - 12M' + S'}{11}$$
$$b = \frac{L' + M' - 2S'}{9},$$

where a encodes a red–green channel, and b encodes a yellow–blue opponent channel. Chroma C is then given by

$$C = 465.5\left(\sqrt{a^2 + b^2}\right)^{0.62}$$

Colorfulness M can be computed from chroma as well as the luminance of the reference white L_W:

$$M = C\big(0.11 \log_{10} L_W + 0.61\big)$$

Finally, the hue angle h is given by:

$$h = \frac{180}{\pi} \tan^{-1} \frac{b}{a}$$

This model is analytically invertible [156], and can thus be used to process images, transforming them from a given scene environment to a specific display environment. Examples of transforming an HDR image to various display conditions are shown in Figure 8.22. Here, all parameters were kept constant, bar the value of E, which was changed according to the type of display device for

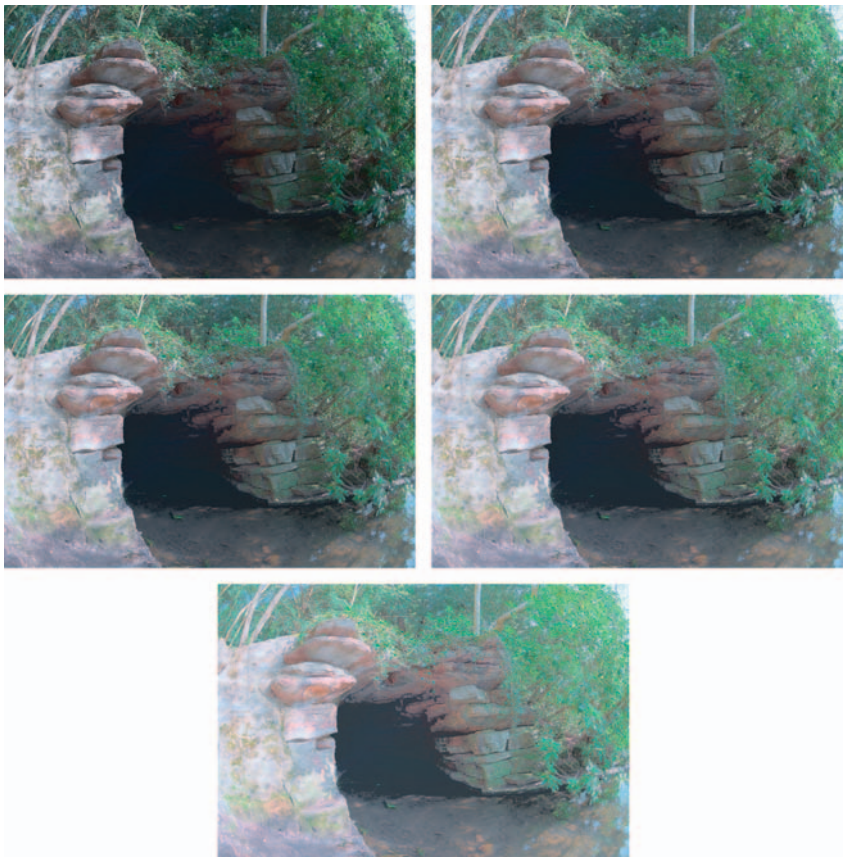


FIGURE 8.22 In reading order, images were prepared for display on an HDR display, high-quality transparency, an LCD display, a CRT, and paper.

FIGURE 8.23 *The output viewing environment was assumed to be 2.5 cd/m^2 in the left image, and 25 cd/m^2 in the right image, for a display device with a peak luminance of 250 cd/m^2.*

which each image was prepared. The input image was assumed to have a white point of $(X_W, Y_W, Z_W) = (28.49, 29.58, 24.28)$ and an input-adapting luminance of $L_a = 0.15\,\text{cd/m}^2$ (computed as the log average luminance). The display environment is characterized in each case with a white point of $(X_W, Y_W, Z_W) = (237.62, 250.0, 272.21)$, and $L_a = 25\,\text{cd/m}^2$.

As the model requires the specification of an adapting luminance for the display environment, the effect of changing illumination can be accounted for, as shown in Figure 8.23. Finally, we show some results for viewing under different illuminants in Figure 8.24.

8.3 OTHER HVS-BASED MODELS

In this section, we discuss several operators based on aspects of human vision other than photoreceptor behavior. They include Tumblin and Rushmeier's operator, which is presented next, the retinex model, Ashikhmin's operator, models based on lightness perception and subband encoding, and finally a model that optimizes a piecewise linear curve based on human contrast perception.

FIGURE 8.24 *In this set of images, the illuminant of the viewing environment is assumed to be (in reading order): CIE A, D50, D65, and D75.*

8.3.1 BRIGHTNESS-PRESERVING OPERATOR

When Miller et al. were the first to introduce computer graphics to the field of lighting design and focused on tone reproduction to accomplish this goal, it was Tumblin and Rushmeier who introduced the problem of tone reproduction to the field of computer graphics in 1993 [326]. Tumblin and Rushmeier also based their work on Stevens's psychophysical data, realizing that the HVS is also solving the dynamic range reduction problem.

The Tumblin–Rushmeier operator exists in two different forms: the original operator [326] and a revised version [327] that corrects a couple of shortcomings, including the fact that it was calibrated in sieverts, which is a unit that is not in wide use. For this reason, we limit our discussion to the revised Tumblin–Rushmeier operator and will refer to it simply as the Tumblin–Rushmeier operator.

Although the revised Tumblin–Rushmeier operator is based on the same psychophysical data as Miller's operator, the brightness function is stated slightly differently:

$$Q(x, y) = C_0 \left(\frac{L(x, y)}{L_\mathrm{a}} \right)^{\gamma}$$

Here, Q is brightness, or perceived luminance, measured in brils. L is luminance in candelas per square meter and L_a is the adaptation luminance, also measured in candelas per square meter. The constant $C_0 = 0.3698$ is introduced to allow the formula to be stated in SI units. Finally, γ is a measure of contrast sensitivity and is itself a function of the adaptation luminance L_a.

This function may be evaluated for an HDR image as well as for the intended display device. This leads to two sets of brightness values as function of input luminances (or world luminances) and display luminances. In the following, the subscripts w and d indicate world quantities (measured or derived from the HDR image) and display quantities.

While Miller et al. conjecture that image and display brightness ratios should be matched, Tumblin and Rushmeier simply equate the image and display brightness values:

$$Q_\mathrm{w}(x, y) = C_0 \left(\frac{L_\mathrm{w}(x, y)}{L_\mathrm{wa}} \right)^{\gamma(L_\mathrm{wa})}$$

$$Q_\mathrm{d}(x, y) = C_0 \left(\frac{L_\mathrm{d}(x, y)}{L_\mathrm{da}} \right)^{\gamma(L_\mathrm{da})}$$

$$Q_\mathrm{w}(x, y) = Q_\mathrm{d}(x, y)$$

The gamma function $\gamma(L)$ models Stevens's human contrast sensitivity for the image as well as the display by plugging in L_{wa} and L_{da}, respectively, and is given by:

$$\gamma(L) = \begin{cases} 2.655 & \text{for } L > 100\,\text{cd/m}^2 \\ 1.855 + 0.4\,\log_{10}(L + 2.3 \cdot 10^{-5}) & \text{otherwise} \end{cases}$$

These equations may be solved for $L_{\text{d}}(x, y)$, the display luminances which is the quantity we wish to display. The result is

$$L_{\text{d}}(x, y) = L_{\text{da}} \left(\frac{L_{\text{w}}(x, y)}{L_{\text{wa}}} \right)^{\gamma(L_{\text{wa}})/\gamma(L_{\text{da}})}$$

The adaptation luminances are L_{da} for the display and L_{wa} for the image. The display adaptation luminance is typically between 30 and 100 cd/m^2, although this number will be higher when HDR display devices are used. The image adaptation luminance is given as the log average luminance L_{wa} (Equation 7.7). The midrange scene luminances now map to midrange display luminances close to L_{da}, which for dim scenes results in a uniform gray appearance of the displayable result. This may be remedied by introducing a scale factor $m(L_{\text{wa}})$, which depends on the world adaptation level L_{da}:

$$m(L_{\text{wa}}) = \left(\sqrt{C_{\text{max}}} \right)^{\gamma_{\text{wd}}-1}$$

$$\gamma_{\text{wd}} = \frac{\gamma_{\text{w}}}{1.855 + 0.4\,\log(L_{\text{da}})}$$

Here, C_{max} is the maximum displayable contrast, which is typically between 30 and 100 for an LDR display device. The full operator is then given by

$$L_{\text{d}}(x, y) = m(L_{\text{wa}}) L_{\text{da}} \left(\frac{L_{\text{w}}(x, y)}{L_{\text{wa}}} \right)^{\gamma(L_{\text{wa}})/\gamma(L_{\text{da}})}$$

For suitably chosen input parameters, a plot of this function is given in Figure 8.25.

As this operator is calibrated in SI units, the image to be tone mapped needs to be specified in SI units also. For an image in unknown units, we experimented with different scale factors before the tone reproduction, and show the results in

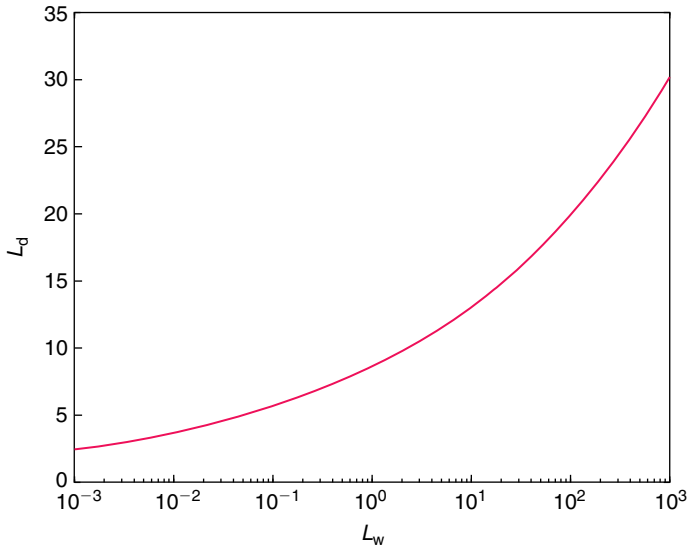


FIGURE 8.25 *Mapping of world luminances to display luminances for the Tumblin–Rushmeier operator.*

Figure 8.26. This image was scaled by factors of 0.1, 1, 10, 100, and 1000, with the scaling resulting in progressively lighter images. For this particular image, a scale factor of close to 1000 would be optimal. Our common practice of normalizing the image, applying gamma correction, and then multiplying by 255 was abandoned for this image sequence because this operator already includes a display gamma correction step.

In summary, the revised Tumblin–Rushmeier tone-production operator is based on the same psychophysical data as Miller's operator, but the crucial difference is that Miller et al. aim to preserve brightness ratios before and after compression, whereas Tumblin–Rushmeier attempt to preserve the brightness values themselves.

FIGURE 8.26 In this sequence of images, the input luminances were scaled by factors of 0.1, 1, 10, 100, and 1000 before applying Tumblin and Rushmeier's revised tone-reproduction operator.

In our opinion, the latter leads to a useful operator which produces plausible results, provided the input image is specified in candelas per square meter. If the image is not specified in candelas per square meter, it should be converted to SI units. In that case, the image may be prescaled by a factor that may be determined by trial and error, as shown in Figure 8.26.

8.3.2 RETINEX

Although Chiu's work is exploratory and is not advertised as a viable tone-reproduction operator, Rahman and Jobson developed their interpretation of the retinex theory for use in various applications, including tone reproduction [142, 260,261]. However, the differences between their approach and Chiu's are relatively minor. They too divide the image by a Gaussian-blurred version with a wide filter kernel.

Their operator comes in two different forms, namely a single-scale and a multi-scale version. In the single-scale version, Chiu's model is followed closely, although the algorithm operates in the log domain. However, the placement of the logarithms is somewhat peculiar, namely after the image is convolved with a Gaussian filter kernel:

$$I_{\mathrm{d}}(x, y) = \exp\Big(\log(I_{\mathrm{w}}(x, y)) - k \log\big(I_{\mathrm{w}}^{\mathrm{blur}}(x, y)\big)\Big)$$

This placement of logarithms is empirically determined to produce visually improved results. We add exponentiation to the results to return to a linear image.

Note that this operator works independently on the red, green, and blue channels, rather than on a single luminance channel. This means that the convolution that produces a Gaussian-blurred image needs to be repeated three times per image.

In the multiscale retinex version, the above equation is repeated several times for Gaussians with different kernel sizes. This results in a stack of images each blurred by increasing amounts. In the following, an image at level n will be denoted $I_{\mathrm{w},n}^{\mathrm{blur}}$. In the examples we show in this section, we use a stack of six levels and made the smallest Gaussian filter kernel 2 pixels wide. Each successive image is convolved with a Gaussian twice as large as for the previous image in the stack.

The multiscale retinex version is then simply the weighted sum of a set of single-scale retinexed images. The weight that is given to each scale is determined by the

user. We have found that for experimentation, it is convenient to weigh each level by a power function that gives straightforward control over the weights. For an image stack with N levels, the normalized weights are then computed by

$$w_n = \frac{(N-n-1)^f}{\sum_{m=0}^{N}(N-m-1)^f}$$

A family of curves of this function is plotted in Figure 8.27. The user parameter f determines the relative weighting of each of the scales. For equal weighting, f should be set to 0. To give smaller scales more weight, f should be given a positive

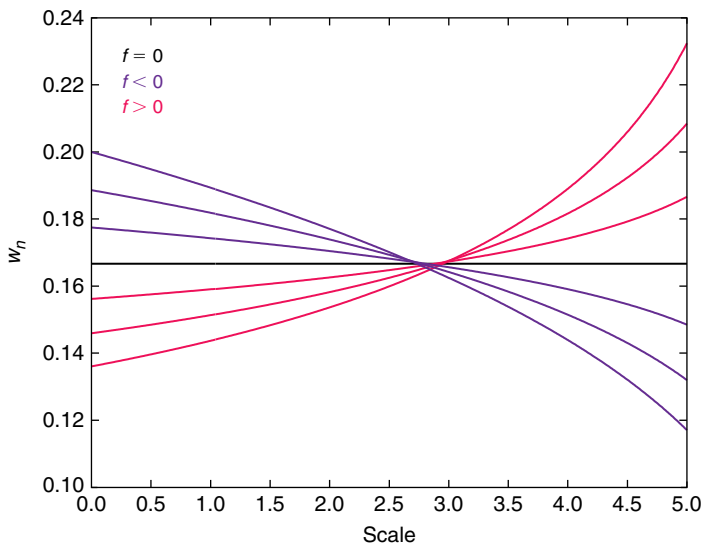


FIGURE 8.27 *Weight factors w_n as function of scale n for different user parameters f. The values for f used range from -0.3 to 0.3 in steps of 0.1.*

value such as 0.1 or 0.2. If the larger scales should be emphasized, then f should be given negative values.

The multiscale retinex takes the following form:

$$I_{\mathrm{d}}(x, y) = \exp\left(\sum_{n=0}^{N} w_n \left(\log(I_{\mathrm{w}}(x, y)) - k \log\left(I_{\mathrm{w},n}^{\mathrm{blur}}(x, y)\right)\right)\right)$$

The two user parameters are k and f, which are in many ways equivalent to the user parameters required to control Chiu's operator. The value of k specifies the relative weight of the blurred image. Larger values of k will cause the compression to be more dramatic, but also create bigger halos. Parameter f, which controls the relative weight of each of the scales, determines which of the Gaussian-blurred images carries most importance. This is more or less equivalent to setting the spatial extent of the Gaussian in Chiu's method. With these two parameters we, therefore, expect to be able to control the operator, trading the amount of compression against severity of the artifacts. This is indeed the case, as Figures 8.28 and 8.29 show.

In summary, Rahman and Jobson's interpretation of Land's retinex theory is similar to the exploratory work by Chiu. There are three main differences. The algorithm works in the log domain, which causes contrasts at large image values to lie closer together. This generally results in fewer issues with haloing. Second, the algorithm operates on the three color channels independently. This approach is routinely followed by various color appearance models (see, for instance, the CIECAM02 model discussed in Section 2.9, and the iCAM model discussed next). Finally, this work operates on multiple scales, which are weighted relative to one another by a user-specified parameter. Multiscale techniques are well known in the literature, including the tone-reproduction literature. Other examples of multiscale techniques are the multiscale observer model, Ashikhmin's operator, and photographic tone reproduction, described in Sections 8.2.1, 8.3.3, and 8.1.2.

8.3.3 ASHIKHMIN'S MODEL

The multiscale observer model aims at completeness in the sense that all steps of human visual processing that are currently understood well enough to be modeled are present in this model. It may, therefore, account for a wide variety of appearance

FIGURE 8.28 *In Rahman's retinex implementation, parameter k controls the relative weight of the Gaussian-blurred image stack. Here, k is varied from 0.0 to 0.25, 0.5, and 0.75 (in reading order).*

effects. One may argue that such completeness is not strictly necessary for the more limited task of dynamic range reduction.

Ashikhmin's operator attempts to model only those aspects of human visual perception that are relevant to dynamic range compression [12]. This results in a significantly simpler computational model consisting of three steps. First, for each point in the image, a local adaptation value $L_{\text{wa}}(x, y)$ is established. Second, a compressive function is applied to reduce the dynamic range of the image. As this step may cause loss of some detail, a final pass reintroduces detail.

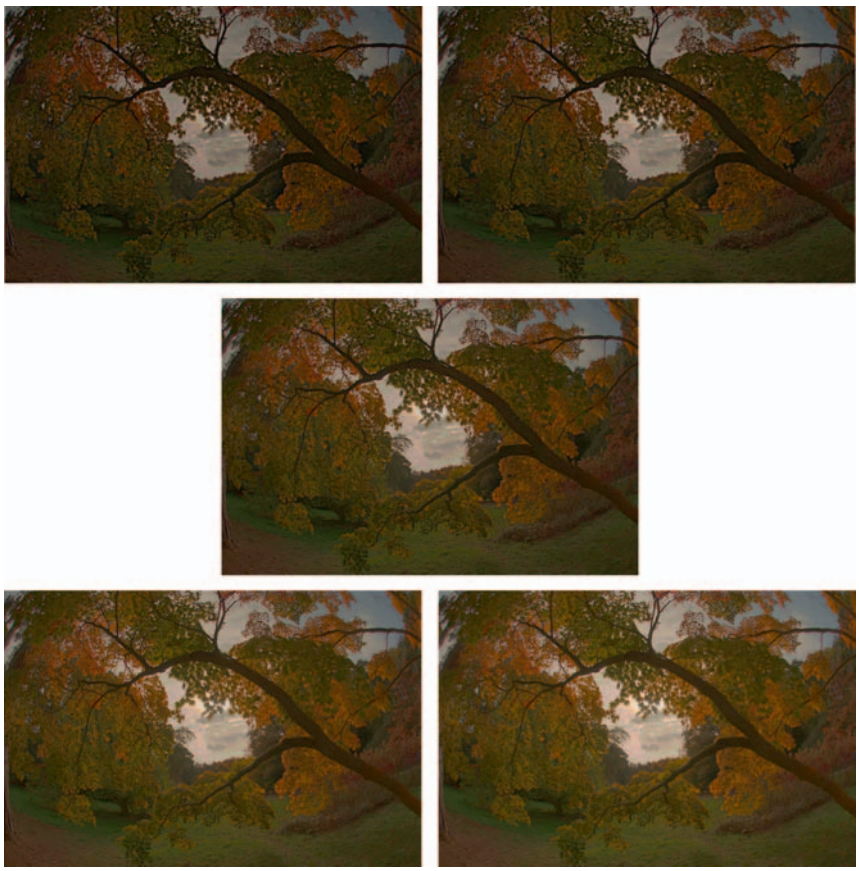


FIGURE 8.29 In Rahman's retinex implementation, the Gaussian-blurred images may be weighted according to scale. The most important scales are selected with user parameter f, which is varied between -4 and 4 in steps of 2. Equal weight is given for a value of $f = 0$, shown in the middle.

Ashikhmin's operator is aimed at preserving local contrast, which is defined as:

$$c_{\mathrm{w}}(x, y) = \frac{L_{\mathrm{w}}(x, y)}{L_{\mathrm{wa}}(x, y)} - 1$$

In this definition, L_{wa} is the world adaptation level for pixel (x, y). The consequence of local contrast preservation is that visible display contrast $c_{\mathrm{d}}(x, y)$, which is a function of display luminance $L_{\mathrm{d}}(x, y)$ and its derived local display adaptation level $L_{\mathrm{da}}(x, y)$, equals $c_{\mathrm{w}}(x, y)$. This equality may be used to derive a function for computing display luminances:

$$\begin{aligned} c_{\mathrm{d}}(x, y) &= c_{\mathrm{w}}(x, y) \\ \frac{L_{\mathrm{d}}(x, y)}{L_{\mathrm{da}}(x, y)} - 1 &= \frac{L_{\mathrm{w}}(x, y)}{L_{\mathrm{wa}}(x, y)} - 1 \\ L_{\mathrm{d}}(x, y) &= L_{\mathrm{da}}(x, y) \frac{L_{\mathrm{w}}(x, y)}{L_{\mathrm{wa}}(x, y)} \end{aligned}$$

The unknown in these equations is the local display adaptation value $L_{\mathrm{da}}(x, y)$. Ashikhmin proposes to compute this value for each pixel from the world adaptation values. Thus, the display adaptation luminances are tone-mapped versions of the world adaptation luminances:

$$L_{\mathrm{da}}(x, y) = F(L_{\mathrm{wa}}(x, y))$$

The complete tone-reproduction operator is then given by

$$L_{\mathrm{d}}(x, y) = F(L_{\mathrm{wa}}(x, y)) \frac{L_{\mathrm{w}}(x, y)}{L_{\mathrm{wa}}(x, y)}$$

There are now two subproblems to be solved, namely, the functional form of tone-mapping function $F(\,)$ needs to be given, and an appropriate local world adaptation level $L_{\mathrm{wa}}(x, y)$ needs to be computed.

To derive the compressive function $F(\,)$, Ashikhmin introduces the notion of perceptual capacity of a range of luminance values. Human sensitivity to luminance changes is given by threshold-versus-intensity functions (see also Chapter 7). This

may be used as a scaling factor for a small range of luminance values ΔL. The intuition behind this approach is that the perceptual importance of a just-noticeable difference is independent of the absolute luminance value for which it is computed. For a range of world luminances between 0 and L perceptual capacity, $C(L)$ may, therefore, be defined as

$$C(L) = \int_0^L \frac{dx}{T(x)},$$

where $T(x)$ is the threshold-versus-intensity function. The perceptual capacity for an arbitrary luminance range from L_1 to L_2 is then $C(L_2) - C(L_1)$. Following others, the threshold-versus-intensity function is approximated by four linear segments (in log–log space), so that the perceptual capacity function becomes

$$C(L) = \begin{cases} L/0.0014 & \text{for} \quad L < 0.0034 \\ 2.4483 + \log_{10}(L/0.0034)/0.4027 & \text{for} \quad 0.0034 \leq L < 1 \\ 16.5630 + (L-1)/0.4027 & \text{for} \quad 1 \leq L < 7.2444 \\ 32.0693 + \log_{10}(L/7.2444)/0.0556 & \text{otherwise} \end{cases}$$

World adaptation luminances may now be mapped to display adaptation luminances such that perceptual world capacity is linearly mapped to a displayable range. Assuming the maximum displayable luminance is given by $L_{\text{d,max}}$, the compressive function $F(L_{\text{wa}}(x, y))$ is given by

$$F(L_{\text{wa}}(x, y)) = L_{\text{d,max}} \frac{C(L_{\text{wa}}(x, y)) - C(L_{\text{w,min}})}{C(L_{\text{w,max}}) - C(L_{\text{w,min}})}$$

In this equation, $L_{\text{w,min}}$ and $L_{\text{w,max}}$ are the minimum and maximum world adaptation luminances.

Finally, the spatially variant world adaptation luminances are computed in a manner akin to Reinhard's dodging-and-burning operator discussed in the following section. The world adaptation luminance of a pixel is a Gaussian-weighted average of pixel values taken over some neighborhood. The success of this method lies in the fact that the neighborhood should be chosen such that the spatial extent of

the Gaussian filter does not cross any major luminance steps. As such, for each pixel, its neighborhood should be chosen to be as large as possible without crossing sharp luminance gradients.

To compute if a pixel neighborhood contains any large gradients, consider a pixel of a Gaussian-filtered image with a filter kernel R of size s as well as the same pixel position of a Gaussian-filtered image with a kernel of size $2s$. Because Gaussian filtering amounts to computing a weighted local average, the two blurred pixels represent local averages of two differently sized neighborhoods. If these two averages are similar, no sharp gradients occurred in the pixel's neighborhood. In other words, if the difference of these two Gaussian-filtered pixels is close to 0, the pixel's neighborhood of size $2s$ is LDR. The DoGs is normalized by one of the Gaussian-filtered images, yielding a measure of band-limited local contrast V_s:

$$V_s = \frac{L_{\mathrm{w}} \otimes R_s - L_{\mathrm{w}} \otimes R_{2s}}{L_{\mathrm{w}} \otimes R_s}$$

The above arguments are valid for any scale s. We may, therefore, compute a stack of band-limited local contrasts for different scales s. The smallest scale is $s = 1$, and each successive scale in Ashikhmin's operator is one pixel larger than the previous. The largest scale is 10 pixels wide.

Each successive larger-scale DoG tests a larger pixel neighborhood. For each pixel, the smallest scale s_t for which $V_{s_t}(x, y)$ exceeds a user-specified threshold t is chosen. By default, the value of this threshold may be chosen to be $t = 0.5$. The choice of threshold has an impact on the visual quality of the operator. If a value of 0.0 is chosen, Ashikhmin's operator defaults to a global operator. If the threshold value is chosen too large, halo artifacts will result. The size of these halos is limited to 10 pixels around any bright features because this is the size of the largest center. To demonstrate the effect of this threshold, we have reduced an image in size before tone mapping, and enlarged the tone-mapped results, which are shown in Figure 8.30.

The size of a locally uniform neighborhood is now given by s_t. The local world adaptation value $L_{\mathrm{wa}}(x, y)$ is a Gaussian-blurred pixel at scale s_t:

$$L_{\mathrm{wa}}(x, y) = (L_{\mathrm{w}} \otimes R_{s_t})(x, y)$$

FIGURE 8.30 *Effect of thresholding on results obtained with Ashikhmin's operator. From left to right: threshold values are 0.0, 0.5, and 1.0.*

Note that the scale s_t will be different for each pixel so that the size of the local neighborhood over which L_{wa} is computed varies according to image content. The idea of using the largest possible filter kernel without crossing large contrast steps is, in some sense, equivalent to the output of edge-preserving smoothing operators such as the bilateral filter discussed in Section 8.5.2.

Other than the threshold value discussed earlier, this operator does not have any user parameters, which is good if plausible results need to be obtained automatically. However, as with several other operators, the input needs to be specified in appropriate SI units. If the image is in arbitrary units, it needs to be prescaled. Figure 8.31 shows an input range of seven orders of magnitude scaled such that the log average luminance is located at $10^{-3}, 10^{-2}, \ldots, 10^{3}$. The output is always scaled between 0 and $L_{\text{d,max}}$, which is $100\,\text{cd/m}^2$ in our case. The response function is notably different dependent on absolute input values, which is because of the capacitance function. Also note that significant clamping occurs for larger values.

In summary, Ashikhmin's operator is based on sufficient knowledge of the human visual system to be effective without aiming for completeness. The operator is not developed to be predictive but to provide a reasonable hands-off approach to producing visually pleasing output in which local contrast is preserved.

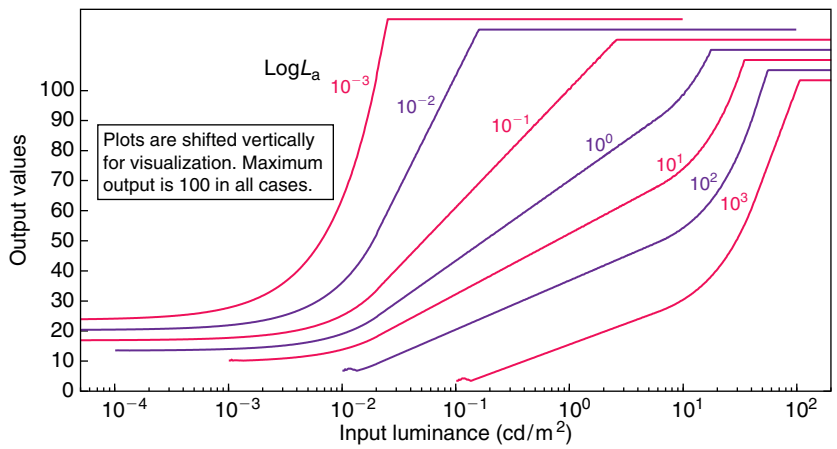


FIGURE 8.31 *The mapping between input and output for Ashikhmin's operator. The input range is always seven orders of magnitude, but shifted such that the log average luminance is as indicated in the figure.*

8.3.4 LIGHTNESS PERCEPTION

The theory of lightness perception provides a model for the perception of surface reflectances (lightness is defined as relative perceived surface reflectance). To cast this into a computational model, the image needs to be automatically decomposed into frameworks, that is, regions of common illumination. As shown in Figure 8.32, lightness perception depends on the overall luminance level. Here, the scene consists of two frameworks. Placing isoluminant probe discs into the scene reveals that their perceived brightness is the same within a frame, but differs across frameworks. The influence of each framework on the total lightness needs to be estimated, and the anchors within each framework must be computed [161,163,165].

It is desirable to assign a probability to each pixel of belonging to a particular framework. This leaves the possibility of a pixel having nonzero participation in multiple frameworks, which is somewhat different from standard segmentation algorithms that assign a pixel to at most one segment. To compute frameworks

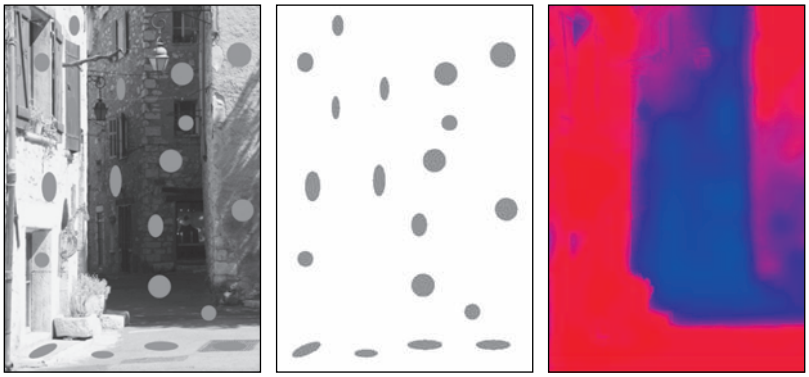


FIGURE 8.32 *The isoluminant probe discs (middle) are placed into a scene (left) with two distinct illumination frameworks (right). The perceived brightness of the discs differs between the frameworks, but is the same within each framework and it is independent of the local surround luminance. The contrast ratios of the discs with respect to local surround in the shadow framework range from 1:2 to 1:9 (disc is an increment) and in the illuminated framework from 9:1 to 2:1 (disc is a decrement). Images courtesy of Grzegorz Krawczyk.*

and probabilities for each pixel, a standard K-means clustering algorithm may be applied.

For each framework, the highest luminance rule may now be applied to find an anchor. This means that within a framework, the pixel with the highest luminance would determine how all the pixels in this framework are likely to be perceived. However, direct application of this rule may result in the selection of a luminance value of a patch that is perceived as self-luminous. As the anchor should be the highest luminance value that is not perceived as self-luminous, selection of the highest luminance value should be preceded by filtering the area of the local framework with a large Gaussian filter.

The anchors for each framework are used to compute the net lightness of the full image. This then constitutes a computational model of lightness perception, which can be extended for the purpose of tone reproduction.

One of the strengths of using a computational model of lightness perception for the purpose of tone reproduction is that, traditionally, difficult phenomena such as the Gelb effect can be handled correctly. The Gelb effect manifests itself when the brightest part of a scene is placed next to an object that is even brighter. Although the formerly brightest object was perceived as white, after the change, this object no longer appears white, but light-gray.

The decomposition of an image into frameworks can be used to build a tone reproduction operator. Figure 8.33 shows examples, along with the frameworks that were derived from the images.

8.3.5 SUBBAND ENCODING

Images can be filtered with band-pass filters, leading to a stack of images each encoding a range of different frequencies. A stack of band-pass images (or subbands)

FIGURE 8.33 *Results of lightness perception tone mapping. The left column shows the original HDR images mapped linearly for display purposes. Their tone mapped versions are shown in the middle column. The right column shows the scene decomposition into frameworks, where red, green and blue colors denote the distinct frameworks. Higher saturation of the color illustrates stronger anchoring within the framework and the intermediate colors depict the influence of more frameworks on an image area. Images courtesy of Grzegorz Krawczyk. The source HDR image in the bottom row courtesy of Byong Mok Oh.*

$V_s(x, y)$ at different scales s can, for example, be created using Equation 8.1. In its simplest form, a band-pass filtered image can be constructed using a straight DoGs scheme:

$$V_s(x, y) = L_s^{\text{blur}} - L_{s+1}^{\text{blur}}, \tag{8.5}$$

where L_s^{blur} is the original image filtered with a Gaussian filter with a kernel size appropriate for scale s. The amount of blur applied at scale $s + 1$ is typically twice that of scale s. The total number of scales $s_{\max}$ can be freely chosen, but would typically not be more than about eight.

It is now possible to split the image $L(x, y)$ into a set of $s_{\max}$ band-pass filtered images. These subbands effectively encode edges at different scales. From this set of images, the original image can be reconstructed by pixel-wise summing of the band-pass images:

$$L'(x, y) = \sum_{s}^{s_{\max}} V_s(x, y) \tag{8.6}$$

For the purpose of tone reproduction, it is possible to apply a gain-control function to each of the subband signals $V_s(x, y)$. A good choice of function would be one whereby stronger edges are attenuated more than weaker edges. This could be achieved by applying sigmoidal compression to the subbands. Li et al. choose the following sigmoidal gain control function $g(V_s(x, y))$ [186]:

$$V_s'(x, y) = g(V_s(x, y)) = V_s(x, y)\,(V_s(x, y) + \varepsilon)^{-2/3} \tag{8.7}$$

Note that for values of $\varepsilon = 0$, this function defaults to a cube root. Applying this function will attenuate large values of $V_s(x, y)$, but at the same time it expands small values of $V_s(x, y)$. As a result, subsequent summing of the subbands $V_s'(x, y)$ will create a distorted image $L'(x, y)$.

Li et al. argue that to reduce visible artifacts, the distortions in the subbands need to be controlled [186]. They analyze the distortions in terms of the effective gain $G_1(x, y)$, defined according to:

$$V_s'(x, y) = V_s(x, y)G_1(x, y) \tag{8.8}$$

The effective gain dips twice over its range of inputs, as shown in Figure 8.34, indicating how and where the distortion in the subbands occurs.

To reduce these distortions, the effective gain $G_1(x, y)$ could be replaced with a smooth activity map $A_s(x, y)$ [186]. The idea is to create an activity map that

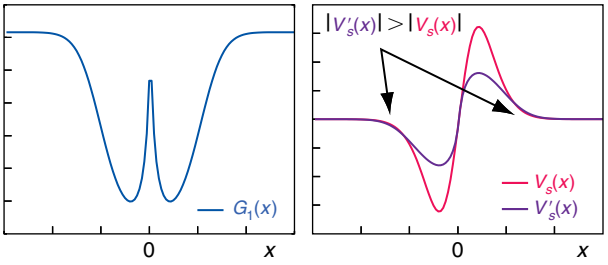


FIGURE 8.34 *The high-frequency behavior of the effective gain (left) distorts the processed subband signals (right), leading to artifacts in the reconstructed image.*

changes more slowly than the subband signal itself, allowing a reduction in the distortions. This can be achieved by taking the absolute value of the subband image and blurring it before applying the gain-control function g():

$$V'_s(x, y) = g(|V_s x, y| \otimes R_s(x, y)) \tag{8.9}$$

$$= g(A_s(x, y)), \tag{8.10}$$

where $R_s(x, y)$ is the Gaussian blur kernel, which will also increase by a factor of two for each subsequent scale s. The remaining question, then, is how to construct an appropriate gain map from the activity map $A_s(x, y)$. Of many possible nonlinear candidate functions, Li et al. choose a gamma-like mapping [186]:

$$G_s(x, y) = \left(\frac{A_s(x, y) + \varepsilon}{\delta_s} \right)^{\gamma - 1}, \tag{8.11}$$

where ε is a parameter modeling the noise floor of the image. The value of $\gamma \in [0, 1]$ controls the amount of compression. The parameter δ_s controls the stability. The gain control is increased for activities larger than δ_s and attenuated for smaller values. To achieve consistency across scales, this parameter can be set as follows:

$$\delta_s = \alpha_s \frac{\sum_{x,y} A_s(x, y)}{MN}, \tag{8.12}$$

where M and N are the width and the height of the subband at scale s. The frequency-related constant α_s is set to 0.1 for the lowest scale, and the 1.0 for scale $s_{\max}$. For intermediate scales, α_s is obtained by linear interpolation.

Finally, these gain maps are used to modify the subband signals:

$$V_s'(x, y) = V_s(x, y)\ G_s(x, y) \tag{8.13}$$

A tone-mapped and artifact-free image can then be reconstructed by summing the filtered subbands, according to Equation 8.6.

Having a different gain map for each scale relies on features in one scale being aligned with corresponding features in other scales. This is generally, but not always, the case in natural images. Thus, a refinement of this operation can be obtained by constructing a single aggregate gain map, rather than using a set of scale-specific gain maps:

$$A_{\text{ag}}(x, y) = \sum_s A_s(x, y) \tag{8.14}$$

$$V_s'(x, y) = m_s A_{\text{ag}}(x, y) V_s(x, y). \tag{8.15}$$

Here, the weights m_s are used to control the relative weighting of each subband.

Finally, the computation of the filter banks themselves can be refined. The simple band-pass filtering discussed earlier in this section has disadvantages when used in conjunction with nonlinear filtering, as required in a tone-reproduction application. In particular, the filtering of a subband may introduce frequencies outside its range. This is another source of error, which can be avoided by using a more sophisticated filter bank.

In this case, analysis–synthesis filter banks are appropriate. In such filter banks, the signal is split by a set of analysis filters, as before. After applying nonlinear filtering to each subband, but before reconstructing the final signal, a second set of filters is applied to remove frequencies outside the range associated with each subband.

Such analysis–synthesis filter banks are normally designed symmetrically, which means that the analysis filters and synthesis filters have the same functional form. Moreover, analysis–synthesis filter banks can be combined with hierarchical subsampling, which reduces the spatial resolution of each subsequent subband by a

factor of two in each dimension. The analysis filters apply subsampling, whereas the synthesis filters incorporate upsampling. An example of such a scheme is used by Li et al., who use Haar wavelets as their filter banks. Although much more refined filters are available, they show that because of the smoothing applied to their gain maps, the resulting tone-reproduction operator is surprisingly effective.

Example images of the resulting tone-reproduction operator are given in Figure 8.35. These images show that very good results can be obtained in this manner. Artifacts such as halos are well controlled, whereas the amount of compression that can be achieved with this operator is very good.

8.3.6 DISPLAY-ADAPTIVE TONE REPRODUCTION

After tone reproduction and requantization, it is inevitable that some parts of the image have undergone distortions with respect to the original HDR image. Mantiuk et al. argue that a good solution to tone reproduction involves keeping these distortions below visible threshold to the extent possible [202]. They, therefore, reformulate tone reproduction as an optimization problem where the objective is to

FIGURE 8.35 *Two challenging example images tone mapped with the scale-space technique proposed by Li et al. [186].*

minimize visible distortions on the one hand, and fitting the image to a particular display setup on the other. The premise is that a tone-mapped image elicits a visual response R_{disp} that ought to match the visual response R_{orig} evoked by the original scene. As a result, a model of human vision can be used as an oracle to judge the quality of a particular tone-mapped image. It can, therefore, be used in an optimization setting as part of the objective function.

The control flow of the resulting tone-reproduction operator is shown in Figure 8.36. The individual models are described in the following, beginning with the tone-reproduction operator. This operator requires parameters that can be optimized for a given input image and a given display configuration. Mantiuk et al. choose to use a piecewise linear function.

The image enhancement module is optional and can, for instance, be used to enhance contrast.

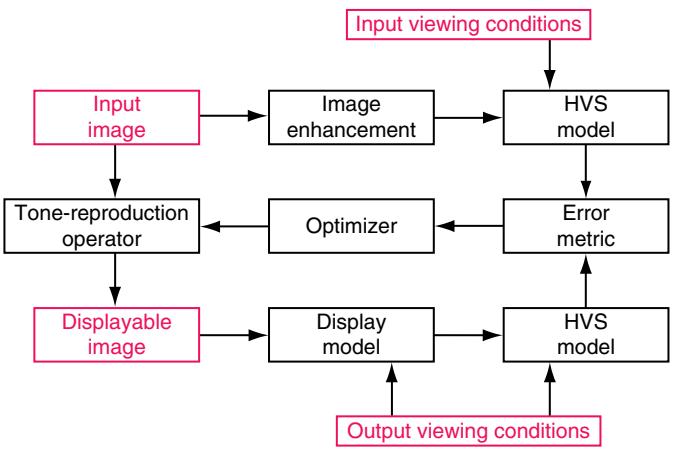


FIGURE 8.36 *Control flow of display-adaptive tone reproduction (after [202]).*

The display model takes into account the capabilities of the display device as well as the viewing conditions. It is argued that room illumination elevates the black level of the display. This approach can, therefore, be seen as a rudimentary description of the display environment. For pixel values $p \in [0, 1]$, the emitted luminance L_{d} can be modeled with

$$L_{\mathrm{d}} = p^{\gamma}(L_{\max} - L_{\mathrm{black}}) + L_{\mathrm{black}} + L_{\mathrm{refl}}, \tag{8.16}$$

where the display range is given by $[L_{\mathrm{black}}, L_{\max}]$ (assuming no light is reflected off the screen), and L_{refl} is the amount of reflected light, which in the case of diffuse screens can be modeled as

$$L_{\mathrm{refl}} = \frac{k}{\pi} E_{\mathrm{amb}}$$

Here, E_{amb} is the ambient illuminance (measured in lux) and k is the reflectivity of the display panel. This reflectivity can be assumed to be around 1% for LCD displays.

The module which implements the model of human vision (HVS in Figure 8.36) is chosen to measure perceived contrast. The response R to an image area with some contrast can be modeled with a transducer function $T(W, S)$ [362], which depends on a measure of local contrast W and human sensitivity to contrast S.

Local contrast can be estimated using a Gaussian pyramid [29], which in this case is computed in log (base 10) space. A DoGs scheme V_s then corresponds to contrast measured over a region related to the level s of the pyramid:

$$V_s = \log_{10} L_s^{\mathrm{blur}} - \log_{10} L_{s+1}^{\mathrm{blur}}$$

To compute contrast in linear space, the result is exponentiated:

$$W_s = 10^{|V_s|} - 1,$$

where W_s is a Weber fraction at scale s.

The sensitivity S_s at scale s can be computed using a standard contrast sensitivity function (CSF) [49] (simplified here by substituting constants where possible):

$$S_s = 250 \min\left(S_1\left(\frac{\rho_s}{0.856 v_d^{0.14}} \right), S_1(\rho_s) \right)$$

$$S_1(\rho_s) = 0.9 \left(1 + \left(3.23 \rho_s^{-0.6} \right)^5 \right)^{-0.2} A \, \rho_s \, e^{-0.9 B \rho_s} \sqrt{1 + 0.06 e^{0.9 B \rho_s}}$$

$$A = 0.801 \left(1 + \frac{0.7}{L_a} \right)^{-0.2}$$

$$B = 0.3 \left(1 + \frac{100}{L_a} \right)^{0.15}$$

Thus, S is a function of the spatial frequency ρ_s (in cycles per visual degree), light adaptation level L_a (in candelas per square meter), and viewing distance v_d (in meters). For a given pyramid level s, the spatial frequency ρ_s can be computed with

$$\rho_s = \frac{\pi h_{\text{res}}}{2^s \, 360 \tan^{-1}\left(\frac{1}{2 n_h} \right)},$$

where n_h is the viewing distance expressed in multiples of the height of the display ($n_h \approx 2$ for typical viewing conditions).

Given Weber fraction W_s and sensitivity S_s, the transducer function is given by [362]:

$$T_s(W_s, S_s) = 12.66 \, \frac{\left(1 + (S_s \, W_s)^3 \right)^{1/3} - 1}{(3.433 + S_s \, W_s)^{0.8}}$$

Thus, the response of the HVS to a contrast W_s, given adaptation level L_a and viewing distance v_d is given by the sum of the responses at all levels s:

$$R = \sum_s T_s(W_s, S_s)$$

The optimization module requires a large number of evaluations of the above transducer function. To make this process tractable, Mantiuk et al. propose to use a statistical measure based on local contrast V_s and luminance $\log_{10} L_{s+1}$. To this end, the dynamic range of the logged input image is divided into N bins (spaced 0.1 log units apart) with centers x_i, with $i \in [0, N)$. The conditional probability density function c is then given by

$$c_{i,s} = P\big((m-0.5)\delta \le V_s < (m+0.5)\delta \| x_i - 0.5\delta \le \log_{10} L_{s+1} < x_i + 0.5\delta\big)$$

with $\delta = x_{i+1} - x_i$ the distance between the bin centers, $m = \{-M, \ldots, -1, 1, \ldots, M\}$, and M chosen such that $M\delta < 0.7$. This function computes for each scale s and each bin i the probability that a particular contrast V_s occurs around a local background luminance x_i. This probability helps in the design of the optimizer's objective function, which is described next.

The objective function, the core of the optimization module, is then constructed by defining the tone curve as a piecewise linear function with end points (x_i, y_i); that is, the values x_i represent input luminances, with corresponding display luminances y_i. Given that the HVS was modeled in terms of contrasts, the objective function is specified in terms of gradients (which encode local contrasts), $d_i = y_{i+1} - y_i$. Assuming the display range is $[L_{\mathrm{d,min}}, L_{\mathrm{d,max}}]$, the tone curve is specified by coordinate pairs (x_i, d_i).

The minimum visual error is then obtained if the human contrast response evoked by the tone-mapped image is as close as possible to the perceived contrast because of the original scene, leading to the following objective function:

$$O = \mathrm{argmin}_{d_1,\ldots,d_{N-1}} \sum_s \sum_{i=1}^{N-1} \sum_{m=-M}^{M} \sum_{m\neq 0} \left(T\left(\sum_{k\in\Phi} d_k, S_{s,d} \right) \right.$$

$$\left. - T\left(e \sum_{k\in\Phi} \delta, S_{s,r} \right) \right)^2 c_{i,s}$$

with the additional constraints that the curve is nonnegative and that the display range is not exceeded:

$$d_i \geq 0$$

$$\sum_{i=1}^{N-1} d_i \leq L_{\text{d,max}} - L_{\text{d,min}},$$

where Φ is given by

$$\Phi = \begin{cases} i+m, \ldots, i-1 & \text{if } m < 0 \\ i, \ldots, i+m-1 & \text{if } m \geq 0 \end{cases}$$

The sensitivities $S_{s,d}$ are related to the viewing environment, whereas $S_{s,r}$ is related to the environment captured by the input image (r refers to the reference image).

After optimization, the piecewise linear tone-curve is given by (x_i, y_i), with y_i specified by:

$$y_i = log_{10} L_{\text{d,min}} + \sum_{k=1}^{i-1} d_k + \left(L_{\text{d,max}} - L_{\text{d,min}} - \sum_{k=1}^{N-1} d_k \right)$$

Display luminance values are then created by exponentiating:

$$L_{\text{d},i} = 10^{y_i}$$

To determine how to encode pixel values p from display luminance values, $L_{\text{d},i}$ are plugged into the inverse of Equation 8.16.

Results produced by this operator for four different displays and display environments are shown in Figure 8.37. The display capabilities for these four images are enumerated in Table 8.1.

8.4 APPARENT CONTRAST AND BRIGHTNESS ENHANCEMENT

In this section, we discuss techniques to enhance perceived contrast and brightness in the image. This is, in particular, important for displaying HDR images on

FIGURE 8.37 *Image tone mapped with the display-adaptive tone-reproduction operator. The four panels show different simulated display conditions. In reading order: CRT display, LCD display, LCD display in office lighting conditions, LCD display viewed in a dark environment. (Image courtesy of Tania Pouli)*

LDR display devices, which exhibit significant deficit of reproducible contrast and maximum luminance levels with respect to real-world scenes. Obviously, physical contrast adjustment to accommodate the display capabilities is the domain of tone mapping. However, in this section, we refer rather to *perceptual effects* than to *physical effects*, which we can *experience* but cannot *measure physically*. Such enhancement of perceived (apparent) contrast and brightness beyond the physical limits of the display device can be considered as postprocessing with respect to tone mapping, which usually does a good job in best use of available physical contrast

Display	Gamma	Black Point	White Point	Ambient Illumination	Screen Reflectivity
CRT	2.2	1.0	80	60	0.02
LCD	2.2	0.8	200	60	0.01
LCD in typical office	2.2	0.8	100	400	0.01
LCD in dark environment	2.6	0.5	500	10	0.01

TABLE 8.1 *Four display specifications simulated in Figure 8.37.*

and maximum reproducible luminance. Also, by having the access to the original HDR images, it is possible to measure contrast and brightness distortions introduced in the tone-mapped image as perceived by the human observer. The apparent contrast and brightness enhancement can be used to reduce the visibility of such distortions on the perceptual side, as little hope can be expected that further improvements can be achieved by a better manipulation of physical contrast and luminance.

Referring to perceptual methods of contrast and brightness enhancement is justified in the light of psychophysical studies, which clearly indicate the human observer preference toward higher contrast and brighter images. For example, Calabria and Fairchild [32] performed the preference experiments with respect to three types of image manipulation: lightness-contrast, chroma-contrast, and sharpness-contrast. The experiment shows that perceived contrast can be manipulated by changing the slope of transfer function (e.g., through change of gamma power function) over the lightness channel, but also by changing the image chroma and sharpness. It turned out that image preference increases with each of the manipulated contrast increases up to 10–20% with respect to the original, but then decreases when the enhancement is too strong. In another study [189], the preference with respect to edge sharpening has been independently confirmed. The "most eye-pleasing sharpness" has been found for the sharpness enhancement of 2.6 times the just-noticeable sharpness difference. Yoshida et al. [376] investigated the subject preference with respect to tone-mapping setup for 14 simulated displays of varying brightness and

dynamic range (reproducible contrast). When such experiment has been performed without the reference to the real scene, the subjects tended to enhance contrast strongly, even at the cost of clipping a large portion of the darkest and brightest pixels. However, when the task was to achieve the best fidelity with respect to a real-world scene, the tone-mapping strategy changed, and the subject avoided clipping both in the dark and light parts of an image, and extended contrast only slightly. As a vast majority of contrast adjustments are done without a direct reference to the scene, the tendency toward contrast boosting can be expected. Yoshida et al. also investigated the user's preference for displays of varying luminance ranges, and clear preference toward brighter displays has been reported. The upper limit over preferred display luminance has been investigated by Seetzen et al. [284], who experimented with HDR displays and determined that the peak display luminance should be less than 6000–7000 cd/m^2 but due to discomfort glare in dim ambient environments a maximum luminance of 400–1200 cd/m^2 is usually sufficient.

All these experiments confirmed that for typical LDR displays, boosting perceived contrast and brightness leads to preferred image appearance. This is, in particular, desirable when performed on the perceptual on the side, which comes as a pure gain with respect to physical display characteristics. In the following section, we discuss a strategy toward apparent contrast enhancement by edge sharpening through a spatial vision effect called the "Cornsweet illusion." In Section 8.4.2, we discuss brightness enhancement techniques by means of the glare illusion.

8.4.1 CORNSWEET ILLUSION

Image contrast is an important factor in the image perception. Obviously, an optical contrast match between the displayed image and real world is usually impossible because of contrast deficit of typical display devices. Fortunately, when we look at images on a computer screen, we often have an impression of plausible real-world depiction, although luminance and contrast ranges are far lower than in reality. So, the key issue in image reproduction is allocating sufficient contrast for a plausible reproduction of all important image features while preserving overall image structure (see Chapter 3 in [339]). The amount of allocated contrast decides on discrimination and identification of objects depicted in the image, which are important factors in image-quality judgment [140].

Existing tone-mapping operators usually are quite efficient in allocating available dynamic range to reproduce image details (local contrast) while maintaining global contrast impression and preserving overall image appearance. Some such operators explicitly process physical contrast (called also "luminance contrast") [89] and use advanced transducer functions (refer to Section 10.9.2) to linearize such physical contrast as perceived by the human observer [208]. As a result, the allocation of physical contrast to reproduce various scale features in the image and preserving its overall structure is pretty good, and further contrast manipulations may lead to changes in the image appearance, which might not be desirable. For example, local contrast enhancement can be performed only at the expense of losing contrast at higher scales and vice versa, which is a direct consequence of the dynamic range deficit. Figure 8.45(b) shows an example of such global contrast loss due to aggressive local contrast enhancement by a gradient domain algorithm [208]. Figure 8.45(a) shows the opposite example, where the global contrast is better reproduced at the expense of texture details reproduction. Because the physical contrast manipulation reaches the dead end in such conditions, the question arises: Can we somehow enhance the perceived contrast?

Contrast perception is a far more complex phenomenon than accounting for physical contrast [267]. The adaptation of the eye and other factors such as luminance levels at the display and ambient lighting as well as the spatial frequencies of the contrast signal play an important role in this respect, which we discuss in Sections 10.7.3 and 10.10. What is, however, more difficult to model is the influence of spatial vision effects, which arise as a result of intensity and color patterns created by neighboring pixels [267]. In this way, perceived contrast is difficult to quantify in isolation from the actual image content. Here, well-known simultaneous contrast phenomenon can be recalled, where the brightness of patches of the same luminance increases when they are imposed on dark background, and vice versa (Figure 8.38[a]). This creates apparent contrast between regions of the same luminance. Refer also to Figure 2.25, in which simultaneous contrast for chromatic channels is also shown. Another well-known phenomenon is the Mach bands effect (Figure 8.38[b]), when illusory light and dark peaks are perceived on both sides of an edge, which separates smooth luminance gradients. Although the luminance signal across the edge is continuous, there is a discontinuity of its first derivative, which triggers apparent contrast in the form of Mach bands that cannot be registered

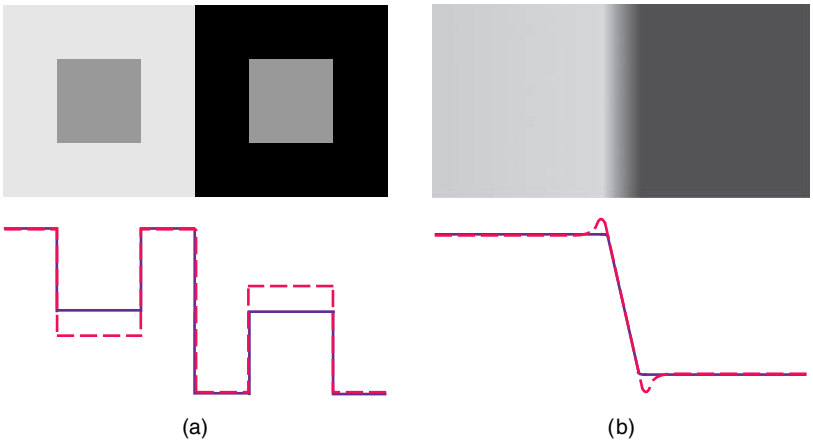


FIGURE 8.38 *Lightness and apparent contrast effects: (a) Simultaneous contrast shown for identical gray patches imposed on the bright and dark background. The purple graph in the bottom plot depicts luminance levels across the horizontal scanline traversing the upper image center. The pink graph depicts the corresponding lightness levels as perceived by the human observer. (b) Mach bands resulting from nonsmooth transition between the luminance gradient (in the center) and flat luminance regions, which are visible as lightness overshooting near the transition regions. The interpretation of the purple and pink graphs is the same as in (a).*

by means of the physical contrast. Although an active employment of simultaneous contrast and Mach bands to enhance apparent contrast in the image in a controllable way seems to be possible, it would likely involve changes in the image structure by introducing out-of-tune spatial features that were not originally present.

In this respect, the Cornsweet perceptual illusion [157] is far more promising, as it enables contrast enhancement along the edges [27]. The illusion is created by introducing a pair of gradient profiles that are gradually darkening and lightening toward their common edge, where they cumulate to a sharp shading discontinuity

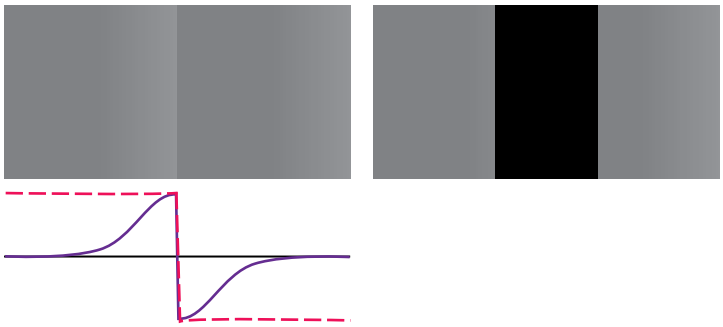


FIGURE 8.39 *Cornsweet illusion: (left) By imposing the so-called Cornsweet luminance profile (as depicted by the purple plot below) onto the center of isoluminant bar, a lightness illusion is created that the left bar side is brighter than the right one (as depicted by the red plot). (right) By occluding the Cornsweet profile with the black rectangle, the lightness of visible bar parts becomes the same, as one could expect for the isoluminant bar.*

as shown in Figure 8.39. Through some filling-in mechanisms of human perception [259] that are not fully understood (perhaps similar to restoring image in the retinal blind spot at the optic nerve exit from the eyeball, or in the foveal blind spot at dark conditions because of the absence of foveal rods), the lightness levels on both sides of the discontinuity are propagated, which creates an impression of the lightness step function. As the gradient profiles (called further the "Cornsweet profiles") gradually tend to the original luminance levels on the both sides of the edge, the apparent contrast is created without any extra use of dynamic range. Note that an alternative way to trigger similar apparent contrast would be to introduce physical contrast as the luminance step function, which obviously requires extra dynamic range. The Cornsweet illusion can also be created by inserting the Cornsweet profile to the uniform luminance region as shown in Figure 8.39, but what is even more interesting is that it can be added to existing edges to boost even

further their apparent contrast. In the latter scenario, skillfully inserted Cornsweet profiles affect relatively little the image appearance as they contribute to the desirable apparent contrast enhancement on the edge, and then gradually vanish without creating any additional sharp-contrast patterns. Obviously, when exaggerated, such profiles can create infamous halo effects as discussed in the context of tone mapping, so an important issue is to control their strength. In the arts, adding such profiles along important profiles and silhouettes is called "countershading" [193]. Figure 8.40 shows a dramatic example where because of countershading, a good

FIGURE 8.40 *In Le Noeud Noir, Georges Seurat uses countershading to enhance the depicted woman's contrast with respect to background.*

separation of the foreground woman from the background has been achieved. Although such countershading does not correspond to any plausible lighting distribution in the scene, it is not perceived as an artifact.

In fact, a whole family of similar profiles, as shown in Figure 8.41, which are often called with the names of their inventors, can trigger the apparent contrast illusion [157]. For example, the one-sided Craik–O'Brien profile shown in the top row in Figure 8.41 is extremely useful to enhance image contrast in the proximity of saturated (clamped) image regions, where there is not enough dynamic range to use fully symmetric Cornsweet profiles. Also, it is remarkable that even more apparent contrast can be induced by cascading Cornsweet profiles as shown in the bottom row in Figure 8.41. This is well visible in Figure 8.42, where the luminance of the small circle in the center is the same as the most external circle, which appears much darker. By changing the polarity of Cornsweet profiles, the opposite effect can be achieved, which means that the apparent contrast polarity can be easily controlled as well.

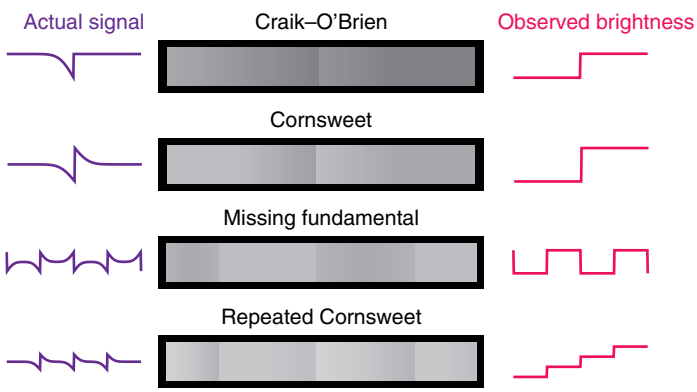


FIGURE 8.41 *Different luminance profiles, which create the Craik–Cornsweet–O'Brien illusion. (Image courtesy of Grzegorz Krawczyk)*

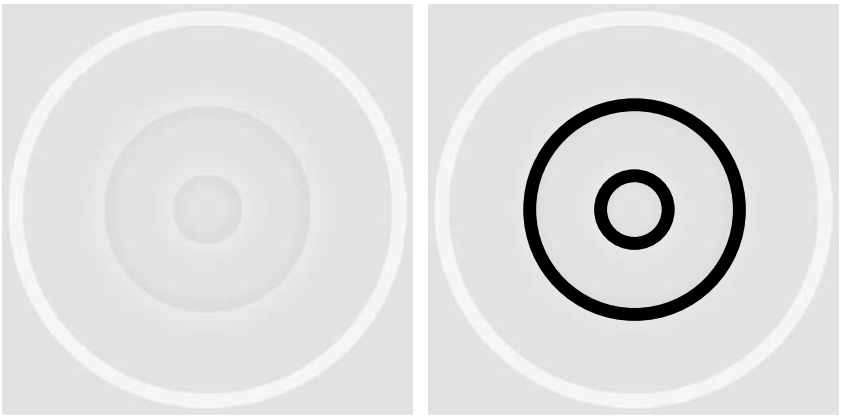


FIGURE 8.42 *Cascading the Cornsweet profiles results in an even stronger illusion. Note that the central circle looks significantly darker than the most external rings (a), while their luminance is the same. By occluding the Cornsweet profiles with the black rings, (b) the lightness of all rings and the central circle appears the same. (Images courtesy of Grzegorz Krawczyk)*

As can be seen in Figures 8.41 and 8.42, the Cornsweet illusion is relatively strong and in optimal conditions can increase the apparent contrast to more than 30%. This is demonstrated in Figure 8.43, which shows an outcome of psychophysical experiment (refer to [68] for more details) in which perceived contrast of the luminance step function has been matched against the Cornsweet luminance profile of varying amplitude (measured as the Michaelson contrast in percentage) and spatial extent (scaled in visual degrees). When the spatial extent tends to infinity, the Cornsweet profile becomes equivalent to the regular step function, which is depicted as the linear dependency. In general, the wider spatial support of the profiles, the stronger the effect. However, excessively wide profiles may interfere

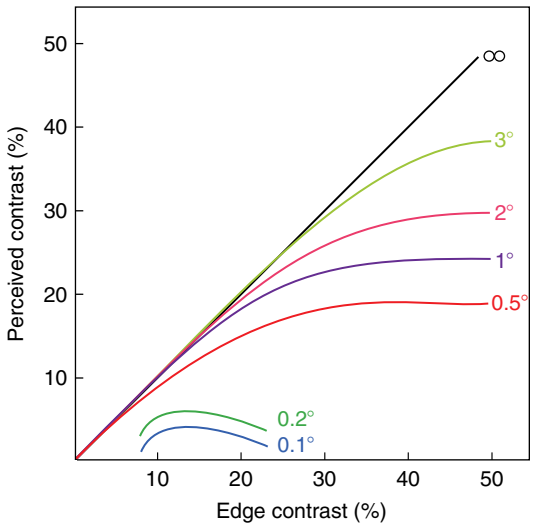


FIGURE 8.43 *Matching apparent contrast resulting from the Cornsweet illusion to the physical contrast resulting from the step function. The Cornsweet profile amplitude as well as the step size are measured as Michelson contrast and expressed in percentage (%). Different plots have been obtained by changing the width of Cornsweet profiles, which is measured in visual degrees. (After [68])*

with the contrast of other edges, which may introduce some visible image artifacts. Thus, the best spatial support selection depends on the image content. Also, excessive increase of the Cornsweet profile amplitude does not lead to equivalent increase in the perceived contrast. The plots saturate for the Cornsweet profile contrast over 30–40%, which depends also on the profile spatial extent. Dooley and Greenfield [68] also discuss the situation where the Cornsweet profile is imposed on existing strong edges, in which case apparent contrast boost is weaker. In what follows, we present a practical algorithm that enables the best selection of the Cornsweet

profile extent and amplitude as a function of contrast lost in tone-mapped image with respect to its HDR counterpart.

At first, let us notice that the Cornsweet profile can be generated by subtracting its low-pass version Y_σ obtained by means of Gaussian blurring from the original image Y. Then, by adding back the difference $U = Y - Y_\sigma$ to Y as

$$Y' = Y + k \cdot U = Y + k \cdot (Y - Y_\sigma) \tag{8.17}$$

the resulting image Y' with enhanced contrast is obtained. This procedure is equivalent to the well-known image-processing technique called "unsharp masking" [257], which is commonly used to sharpen image detail appearance (this increases the apparent contrast as well [32,189]). Thus, unsharp masking is capable of introducing Cornsweet profiles, and the perceptual effect of unsharp masking can be explained by the Cornsweet illusion [296]. Note that by enlarging the size of support σ in a Gaussian filter, more spatial frequencies are added to the original image Y, which effectively enables sharpening of larger image structures. The factor k enables control of the strength of the sharpening effect. Visually, the best sharpening results are obtained for image features whose scale corresponds to the size of σ [238]. Unsharp masking tends to reintroduce high-frequency signal, which may enhance the noise visibility (sensor noise, stochastic noise in rendering) and image-compression errors such as blocky artifacts in JPEG. Also, the Cornsweet effect is significantly weaker at high-contrast edges [157] as, for example, those at the silhouettes of large-scale objects, so adding strong unsharp masking signal U at such edges cannot contribute to any significant contrast enhancement but may cause visible artifacts. To prevent such situations, adaptive gain k (medium contrast enhanced more than strong contrast) and support σ (adapted to detail variance to reduce noise enhancement) have been proposed [265,255].

In this section, we are concerned with a particular application of unsharp masking to restore image details lost because of tone mapping, in which case more selective procedure to build Cornsweet profiles would be desirable. For example, local tone-mapping operators may be pretty good in preserving local details (refer to the texture appearance in Figure 8.45[b]), so a direct application of Equation 8.17 would lead to unnecessary contrast enhancement of such details, while perhaps only global contrast enhancement is needed. A similar case is illustrated in Figure 8.44 (middle),

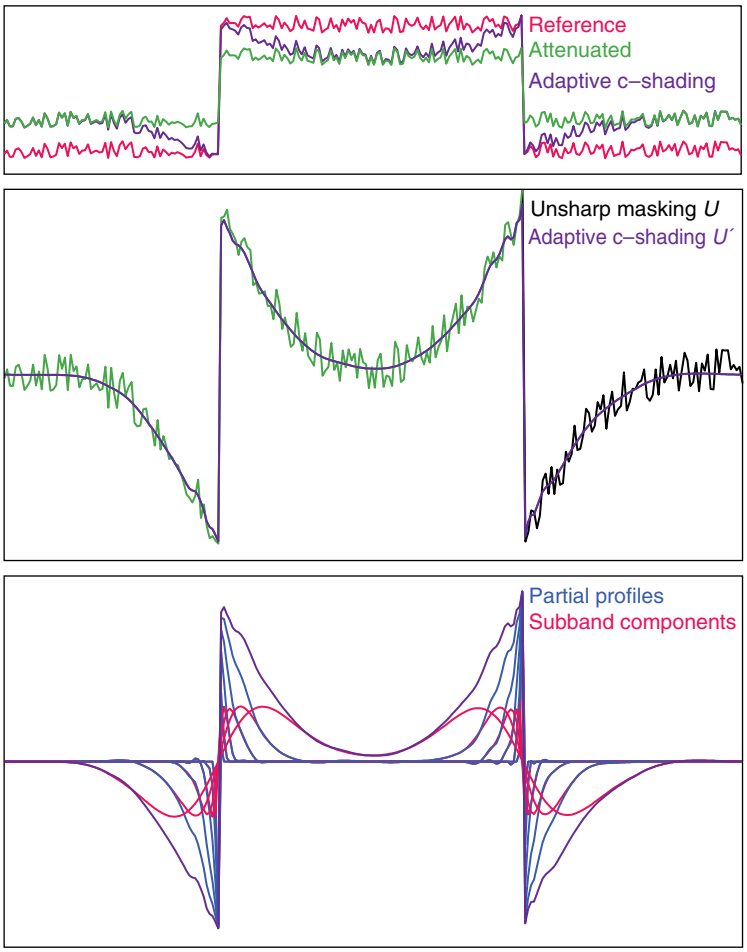


FIGURE 8.44 *Top: Adaptive countershading profile for the reference step edge with details, where the step edge is attenuated while the details are preserved with respect to the reference. Middle: The adaptive countershading method recovers the smooth profile U', which prevents artifacts. The unsharp mask profile U comprise exaggerated details, which are then added for the second time to the input signal. Bottom: The subband components as well as the partial profiles resulting from adding subsequent subband components. (Plots courtesy of Grzegorz Krawczyk [166])*

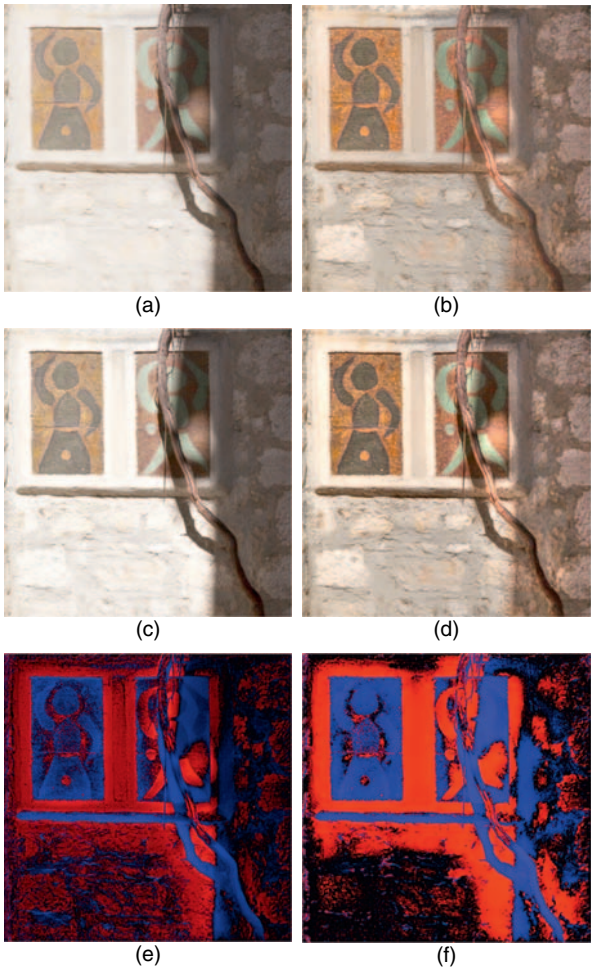


FIGURE 8.45 *Adaptive countershading for tone-mapped images using photographic [274] (left column) and contrast equalization [208] (right column) operators. Tone-mapped images (a) and (b) are shown in the upper row, as well as the outcome of adaptive countershading postprocessing (c) and (d) for these images. The bottom row shows the corresponding countershading profiles (e) and (f), where blue and red colors denote negative and positive polarity of the corrective signal U' (see Equation 8.21). (Images courtesy of Grzegorz Krawczyk)*

where the plot depicts the unsharp masking signal U that causes detail enhancement, which is not required, as tone mapping preserves them well in Figure 8.44 (top). To avoid such problems, Krawczyk et al. [166] propose a multiresolution metric of local contrast, which detects feature loss in tone-mapped image with respect to HDR reference, and drives the spatial extent and the strength of Cornsweet profiles. Krawczyk et al. call their algorithm "adaptive countershading." At first, the HDR and LDR images are converted into a Gaussian pyramid representation, and for each pyramid level i, the local subband contrast C_i is computed as

$$C_i = \frac{|Y_i - Y_{i-1}|}{Y_{i-1}}, \tag{8.18}$$

where Y_i denotes the luminance of a pixel at the pyramid level i, and Y_{i-1} the corresponding low-pass filtered pixel at the lower pyramid level $i-1$. Then, the local contrast distortion measure R_i between the LDR and HDR images for the pyramid level i is

$$R_i = \min\left(1, \frac{C_i^{\text{LDR}}}{C_i^{\text{HDR}}}\right), \tag{8.19}$$

where clamping to 1 (no distortion) is performed in case the details are amplified with respect to the HDR reference, so that no contrast enhancement is needed anyway. Now, the distortion measure R_i is available locally and is used to selectively add countershading profiles in the image regions and for spatial frequency bands, where contrast enhancement is needed. Notice that the addition of successive levels of the full-resolution DoGs up to a certain level (here we present the example for three levels)

$$U = (Y - Y_{\sigma(1)}) + (Y_{\sigma(1)} - Y_{\sigma(2)}) + (Y_{\sigma(2)} - Y_{\sigma(3)}), \tag{8.20}$$

gives the same result as unsharp mask, Equation 8.17, for this level: $U = (Y - Y_{\sigma(3)})$, where $\sigma(i) = 2^{i-1}/\sqrt{2}$ denotes the Gaussian blur at the level i and Y is the luminance of the reference image. Note also that the DoG is equivalent to the band-pass decomposition of an HDR image, where each band contains signal of similar spatial frequencies. By adding such band-pass information back to the LDR image with the gain factor $(1 - R_i)$, which is proportional to the measured contrast ratio distortion (refer to Equation 8.19), the lost contrast information can be enhanced

for this particular band. By summing up such restoration components for all N bands

$$U' = \sum_{i=1}^{N} (1 - \uparrow R_i) \cdot \left(\log_{10} Y^{\mathrm{HDR}}_{\sigma(i-1)} - \log_{10} Y^{\mathrm{HDR}}_{\sigma(i)} \right), \tag{8.21}$$

we obtain the countershading profiles U', which are adjusted to match the contrasts in the reference image. Here, it is assumed that the contrast ratios R_i are upsampled to the full resolution as denoted by the ($\uparrow$) operator, while the low-pass luminance images $Y^{\mathrm{HDR}}_{\sigma(i)}$ are computed in the full image resolution to preserve the phase information, which would otherwise be lost by the resolution reduction. The luminance Y^{HDR} and the countershading profiles U' are calculated in the logarithmic space, which approximates perceptual response to luminance and prevents too-strong darkening inherent for the computation performed in the linear luminance space. The contrasts in the tone-mapped image are restored by adding the countershading profiles U' to the luminance of the input image in the logarithmic space. Figure 8.44 illustrates this process for a simple step function, whose contrast has been attenuated because of detail-preserving tone mapping.

Finally, excessive countershading profiles may become visible as halo artifacts and degrade the image quality, which in most cases is unacceptable and in fact reduces the strength of the contrast enhancement (refer to Figure 8.43). An perception-inspired visual detection model can be used to estimate the maximum amplitude of Cornsweet profile that is not objectionable in a given area. Krawczyk et al. [166] propose such a model, which determines tolerable amplitudes of the countershading profiles by applying a customized CSF that is tailored specifically for the Cornsweet illusion [68]. Refer to Section 10.7, where examples of similar perceptual models, which include such HVS characteristics as luminance masking, CSF, and contrast masking (visual masking), are treated in more detail.

Figure 8.45 illustrates the outcome of adaptive countershading for an HDR image processed with the photographic [274] and contrast equalization [208] tone-mapping algorithms. In the former case (left column), the visibility of texture details has been enhanced. For the contrast equalization operator (right column), countershading enhances the shadow boundaries and restores brightness relations between the image areas, which have been suppressed because of aggressive texture detail enhancement by this operator.

So far, we discussed the Cornsweet illusion in the context of luminance signal both for measuring lost contrast with respect to the reference HDR image as well as for inserting the corrective profile U' into the image. However, one can imagine that the contrast lost is still measured for luminance, while the corrective signal U' modifies other pixel components such as hue or chroma. Because the hue manipulation may lead to unnatural image appearance, chroma scaling is more promising in this respect, as it is commonly performed in photography. Chroma enhancement strengthens image colorfulness, and it turns out that increasing the original image chroma by 10–20% usually results in preferred image appearance [90]. Although this may lead to less-natural colors with respect to the real world, the human color memory is unreliable and preference toward more colorful images is quite sustained. Smith et al. [297] experimented with introducing the Cornsweet profile into the chroma signal to enhance overall image contrast. Figure 8.46 shows a successful example, where colorfulness contrast introduced by means of the Cornsweet profile at the border between the sky and buildings creates an impression of greater dynamic range, promotes separation of foreground and background objects, and improves the sense of depth in the image.

FIGURE 8.46 *Global contrast enhancement in the original tone-mapped image (a) by introducing Cornsweet profiles into the chroma channel in (b). (Images courtesy of Kaleigh Smith and Grzegorz Krawczyk [297] © 2007 Eurographics Association 2007. Used with permission.)*

Countershading proved also to be useful in other applications such as color-to-gray image conversion, where the dimensionality reduction from usual three color channels into one achromatic channel may lead to information loss. As shown in [298], lost chromatic contrast can be restored by adding the Cornsweet profiles into the achromatic channel.

In three-dimensional (3D) graphics, other signals can be used to derive countershading profiles, such as image depth [197]. Also, as discovered by Purves et al. [259], the apparent contrast enhancement is significantly stronger when Cornsweet profiles are consistent with the scene lighting. The effect is further enhanced if perspective foreshortening and other cues resulting from 3D scene interpretation are present. Figure 8.47 shows an example where the apparent contrast enhancement with respect to a standard two-dimensional (2D) Cornsweet profile increased 167%, as reported in [259]. Motivated by this observation, Ritschel et al. [277] proposed a *3D unsharp masking* technique, where the Cornsweet profiles are created directly in the object space over outgoing radiance function. This way, the depiction of

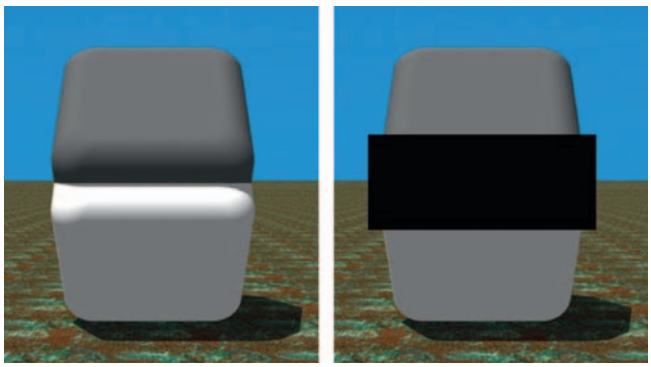


FIGURE 8.47 *The Cornsweet effect strength in 3D scenes. Note how different is brightness of the bottom cube's wall in both images, while they differ only in the presence of black rectangle, which occludes the Cornsweet profile in the right image. (Images courtesy of Dale Purves [259]; Image by R. Beau Lotto at www.lottolab.org with Dale Purves at www.purveslab.net)*

FIGURE 8.48 3D unsharp masking: (left) original image, (middle) enhanced image, and (right) the corresponding correction signal U' but computed and added to the surface shading in the 3D object space. (Images courtesy of Kaleigh Smith and Tobias Ritschel [298])

various visual cues has been enhanced, including gradients from surface shading, surface reflectance, shadows, and highlights, which improves overall image appearance and eases the 3D scene interpretation. Figure 8.48 shows an example of an image enhanced using 3D unsharp masking, where the Cornsweet profile inserted at the shadow boundary deepens the shadow blackness impression and, at the same time, enough dynamic range in the shadow is left (it is not completely black) so that the text on the opened book page remains readable.

8.4.2 GLARE ILLUSION

Every real-world optical system is affected by a certain amount of reflected (at lens and sensor surfaces, as well as the lens mount) and scattered (diffusion in lenses) stray light [266]. This results in the contrast reduction because of a veil of luminance in the proximity of bright light sources and highlights. This unwanted effect is called *veiling glare* or simply *glare*, and it is, in particular, a problem in HDR imaging where direct capturing of such bright objects is very common. Another related effect referred to as *lens flare* creates sharper artifacts such as ghost images of the lens aperture and diffraction streaks because of the aperture leaves, which for spatially extended light sources become blurred and contribute to the veiling glare as well [316]. McCann and Rizzi [215] have measured glare effect specifically in multiexposure HDR imaging, and showed that it can

significantly reduce the dynamic range of captured photographs because of overall contrast reduction. In Section 5.10, we discussed the lens flare removal in HDR images, through a 2D deconvolution operation based on the estimated camera's point spread function (PSF), which is a physical measure of system optics. Talvala et al. [316] propose a novel HDR image-acquisition method, where in front of photographed scene they place a structured occlusion mask, which blocks the light contributing to glare for selected pixels. As a result, occluded pixels capture just the glare information, which can be then subtracted from nonoccluded pixel intensity values.

In this section, we are concerned with the biological optical systems such as the human eye, which are even more prone for glare effects because of structural shape imperfections of optical elements as well as the presence of various particles suspended in the lens and vitreous humor (refer to Figure 8.52). In the context of human vision, the term "disability glare" is often used to denote the reduction of visual performance in the presence of bright objects in the field of view. Another term, "discomfort glare," refers to a different situation where the overall illumination is too high, for example, sunlight reflected by the snow. In this section, we investigate synthetic rendering of typical glare patterns in LDR images, which are somehow interpreted by the human brain as caused by the real presence of bright objects and lead to *glare illusion*.

The levels of contrast required to naturally trigger the glare effects in the eye cannot be achieved using traditional LDR display technology; so, to improve image realism, glare patterns are often painted directly in the images. This is, in particular, important in realistic image synthesis [303], flight and drive simulators [235], and computer games [150]. Moreover, it turns out that when a veiling pattern is painted around nominally bright objects in the image, they appear brighter because of the glare illusion, and the perceptual effect of image dynamic range expansion can be observed. This is illustrated in Figure 8.49, where the sun and its nearest surround appear much brighter than the white circle, although its pixel intensities are the same as for the sun center. The effect of glowing caused by introducing smooth gradients around bright objects can be easily obtained [378,379] even for abstract scenes without any semantic relation to the sun or other bright light sources, as illustrated in Figure 8.50. Such gradients have been used by artists for centuries to improve apparent dynamic range of their paintings, and it is just as attractive today in a digital imaging context. A recent perceptual study [375] demonstrates

FIGURE 8.49 *A snapshot from a video game, Alan Wake, being developed by Remedy Entertainment with luminance gradients added around the sun to enhance its perceived brightness. Such effects cannot be seen for the white circle inpainted on the right, although the same pixel intensity has been set as for the central part of the sun.* © *Copyright 2005–2010 Remedy Entertainment Ltd. All rights reserved.*

that the impression of displayed image brightness can be increased by more than 20% by convolving high-intensity pixels in the image with relatively simple filters [303,150], whose role is essentially to blur image details around bright objects. This requires precise information on the pixel intensity, which is readily available in HDR frames but might be difficult to reconstruct in saturated regions in LDR frames. Meaningful recovery of light and highlight intensities in LDR frames is one of the main difficulties in the so-called inverse tone-mapping problem (see Chapter 9).

A typical glare pattern for a small light source as perceived by most subjects with normal eyes is depicted in Figure 8.51. Two major components can be distinguished in the glare pattern [294,303]: (1) *bloom*, a general loss of contrast in the proximity

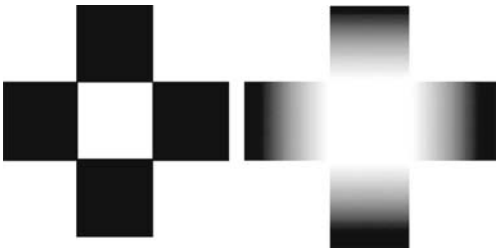


FIGURE 8.50 *It is not necessary to have physical light sources to evoke the impression that an object is self-luminous, as can be seen in the right image. Painting halo around objects enhances their brightness or even creates an impression of glow without the actual light emission. Recent studies indicate that the perception of luminosity is strongly linked to luminance gradients that accompany surface representations [151]. Interestingly, the same surface representations (refer to the image on the left) without such luminance gradients cannot trigger the self-luminosity impression, and instead are perceived as reflective even when displayed on a light-emitting computer screen. (Redrawn from [379])*

of bright objects (veil) and (2) *flare*, which comprises the *ciliary corona* (the sharp needles) and the *lenticular halo*. Some people report temporal changes in the glare appearance such as a pulsation of the glare intensity and flickering of the fine needles in the ciliary corona [276].

The majority of existing approaches to computer-generated glare, while inspired by knowledge about human eye anatomy and physiology, are based on phenomenological results rather than explicit modeling of the underlying physical mechanisms. A common approach is to design convolution filters, which reduce image contrast in the proximity of glare sources by, effectively, adding luminance gradients around bright regions patterns similar to that shown in Figures 8.49 and 8.50. Spencer et al. [303] base their filter on the PSF (or its analog in frequency domain, called optical transfer function [OTF]) measured for the optics of the human eye [358,64, 210,20]. (Refer also to Section 10.7.1, where we describe in more detail the OTF

FIGURE 8.51 *The glare appearance example. (Image courtesy of Tobias Ritschel [276])*

model proposed by Deeley et al. [64] in the context of visible differences prediction between images.) Glare solutions used in tone mapping [347,73] are mostly based on Spencer et al.'s approach. A set of Gaussian filters with different spatial extent, when skillfully applied, may lead to very convincing visual results. This approach is commonly used in computer games [150] and postproduction rendering, as shown in Figure 11.4 (Section 11.2.5). According to perceptual studies by Yoshida et al. [375], a simple Gaussian filter leads to a similar brightness enhancement as a more complex PSF inspired by human eye optics. Other glare effects such as the ciliary corona and the lenticular halo are often designed off-line, and placed in the location of the brightest pixel for each glare source as a billboard (image sprite)[278,303]. In the designing of such billboards, seminal opthalmology references are used, such as in [294]. The resulting appearance is very realistic for small point-like glare sources.

However, using billboards, it is difficult to realistically render glare for glare sources of arbitrary shape and nonnegligible spatial extent.

Recently, Kakimoto et al. [146], van den Berg et al. [330], and Ritschel et al. [276] investigated the application of wave optics principles to glare modeling. They considered various obstacles causing light scattering on its way from the pupil to the retina, as shown in Figure 8.52. In particular, the eyelashes, the eyelids, and the pupil edge have been identified as the main reason for light diffraction in the eye by Kakimoto et al. Van den Berg investigated the role of small particles randomly distributed in the lens and floating in the vitreous humor in creating the ciliary

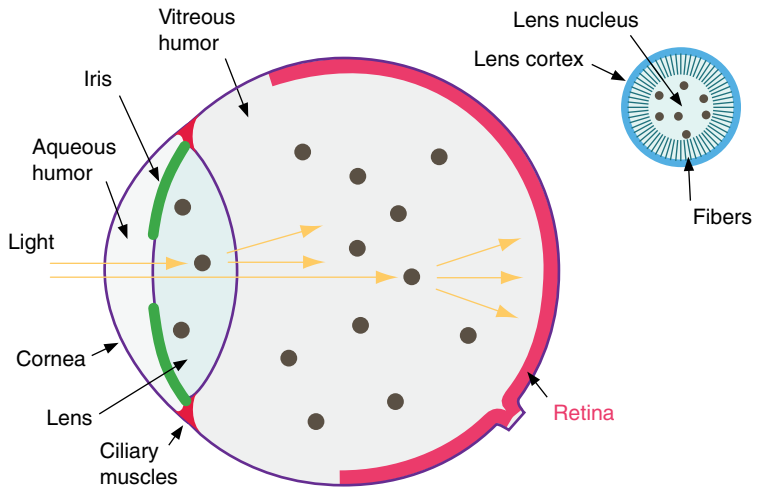


FIGURE 8.52 *The eye anatomy. The upper-right inset shows the lens structure with fiber membranes in the lens cortex, which act as a diffraction net and are responsible for the lenticular halo effect. Notice also opaque particles suspended in the lens and vitreous humor, which create the ciliary corona pattern because of light scattering as shown in Figure 8.51. (Image courtesy of Tobias Ritschel [276] © The Eurographics Association 2009. Used with permission.)*

corona pattern as originally suggested by Simpson [294]. Regularly spaced fiber membranes in the lens cortex act as the diffraction grating, which produces lenticular halo [294,278,303]. Finally, Ritschel et al. considered the dynamic aspect of glare, observing that the pupil size, the shape of lens, and particle positions change in time (refer to Figure 8.52).

It turns out that under a simplifying assumption that all these opaque light-scattering obstacles are orthographically projected into a single plane aperture (Figure 8.53), a simple wave-optics model can be derived. Essentially, the model boils down to taking the Fourier transform $\mathcal{F}$ of the pupil aperture function $P(x_\mathrm{p}, y_\mathrm{p})$, and computing its squared value $|\mathcal{F}\{P(x_\mathrm{p}, y_\mathrm{p})\}|^2$ to derive the PSF for this aperture. Then, it is enough to convolve the input HDR image with the PSF to

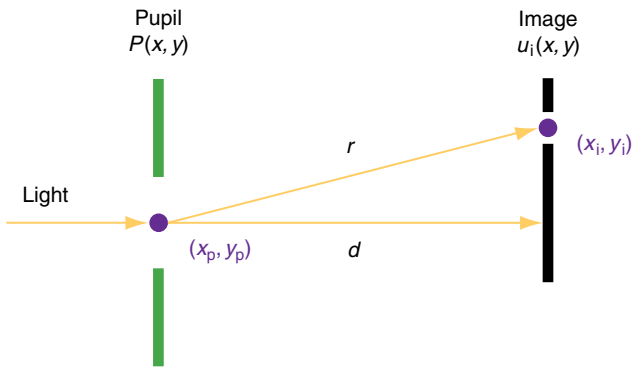


FIGURE 8.53 *A simplified scheme of the optical system in the eye used for wave optics simulation of light scattering in the eye. While placing all opaque light obstacles in the single aperture plane (the pupil plane $P(x_\mathrm{p}, y_\mathrm{p})$) poorly approximates their actual position in the eye (refer to Figure 8.51), the visual glare appearance is similar with respect to properly simulated multiplane aperture, where the position of each plane corresponds to the actual position of each obstacle in the eye [276]. (Image courtesy of Matthias Ihrke [276])*

obtain glare images similar to those in Figure 8.55 (right). More formally, the Fresnel approximation to Huygen's principle [106] for the optical system, as depicted in Figure 8.53 under the assumption of homogeneous incident light of unit amplitude, is given as

$$L_i(x_i, y_i) = K \left| \mathscr{F}\left\{ P(x_p, y_p) E(x_p, y_p) \right\}_{p=\frac{x_i}{\lambda d}, q=\frac{y_i}{\lambda d}} \right|^2 \qquad (8.22)$$

$$K = 1/(\lambda d)^2$$

$$E(x_p, y_p) = e^{i \frac{\pi}{\lambda d}(x_p^2 + y_p^2)}$$

where (x_i, y_i) denote the coordinates at the retina plane, λ is the wavelength of the light, and d the distance between the pupil and retina (refer also to Figure 8.53). The pupil aperture function $P(x_p, y_p)$ gives the opacity of each point in the pupil (0 transparent, 1 opaque). The Fourier transform $\mathscr{F}$ is computed at the grid of points $(p, q) = \left(\frac{x_i}{\lambda d}, \frac{y_i}{\lambda d}\right)$.

Figure 8.54 summarizes the computation process that is performed in real-time using recent graphics hardware. At first, the high-resolution opacity image

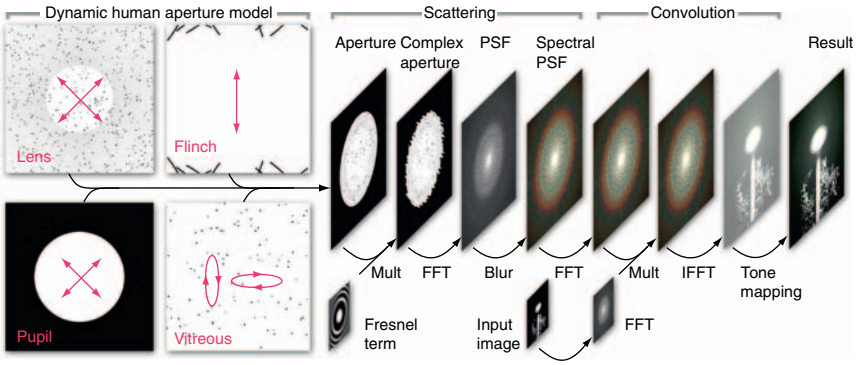


FIGURE 8.54 *The glare computation pipeline, where the black arrows indicate GPU shaders that transform input texture(s) into output textures. (Image courtesy Tobias Ritschel [276] © The Eurographics Association 2009. Used with permission.)*

is composed from the images representing: the pupil aperture, the lens diffraction grid (at the lens cortex) and particles, the eyelashes, and particles in the vitreous humor (obviously, other light-scattering elements can be considered as well). The resulting aperture function $P(x_p, y_p)$ is multiplied by the imaginary Fresnel term $E(x_p, y_p)$ as in Equation 8.22. Then, the Fast Fourier Transform (FFT) is performed over the resulting complex aperture, and the PSF function for a single wavelength is computed. By rescaling the PSF function for different wavelengths (refer to [276] for more details) and summing up all resulting PSFs, the spectral PSF is computed. Then, the input HDR image convolution with the spectral PSF is performed in the Fourier domain, where it boils down to a simple multiplication. The Inverse Fast Fourier Transform (IFFT) is performed to obtain the image with glare in the spatial domain, which is then tone mapped and displayed.

Figure 8.55 (right) shows an example of the candle image, which has been computed using the processing pipeline as shown in Figure 8.54. Note the differences in the visibility of horizontal needles in the ciliary corona with respect to simple billboard approach Figure 8.55 (left), where the spectral PSF is just placed at

FIGURE 8.55 *The glare appearance for a luminous object of nonnegligible spatial size: (left) the billboard approach, where the light-diffraction pattern (point spread function) is just imposed on the background candle image, instead of being convolved with this image (right), which leads to a more physically correct result. (Image courtesy Tobias Ritschel [276] © 2009 Eurographics. Reprinted by permission.)*

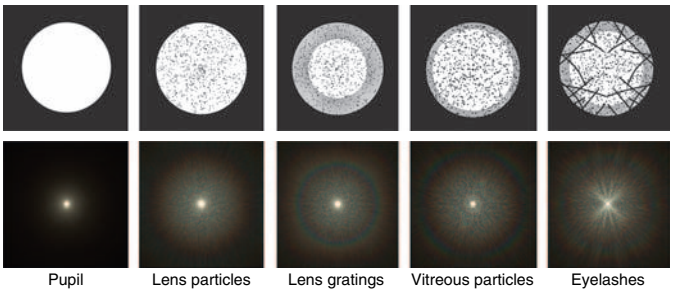


FIGURE 8.56 *Adding light scatterers from left to right. In the upper row, the opacity images, which are used as the aperture function $P(x_p, y_p)$, are shown. In the bottom row, the corresponding spectral PSF are depicted. Note that the spectral PSF for vitreous and lens scattering because of suspended particles looks similar for static glare, and the difference can be observed mostly when dynamics of their motion is simulated, as in [276]. (Image courtesy Tobias Ritschel)*

the flame center, which is a commonly used approach. In fact, pronounced ciliary corona, as in Figure 8.55 (left), is visible only for objects whose angular extent is less than 20 min of arc [294], in which case the billboard approach is fully justified.

Figure 8.56 depicts the impact of adding opaque occluders, which are inherent for different anatomical parts of the eye, on the appearance of the partial PSF that includes only subsets of the pupil aperture images.

8.5 OTHER TONE-REPRODUCTION OPERATORS

Human vision has served as the inspiration for a large number of tone-reproduction operators, on the basis that it solves a similar problem: Some signals in the retina are transduced as voltages, which are strictly bounded between a given minimum and maximum. In the rest of the brain, signals are communicated by means of spike

trains, which have a lowest and highest frequency. The light incident on the retina, however, is virtually unbounded.

However, this does not mean that tone reproduction must be specified in terms of human vision. It is entirely possible to engineer operators without taking human vision as a lead. In this section, several of the most prevalent operators in this class are discussed. Histogram adjustment is a technique that produces images in a fully automatic manner. The importance of both bilateral filtering and gradient domain techniques lies not only in the fact that they can both be used directly as tone-reproduction operators, but also in their ability to serve as general techniques to help solve other problems. For instance, the decomposition into base and detail layers by means of bilateral filtering was already encountered in the description of iCAM06 in Section 8.2.3. Moreover, bilateral filtering can be used as a technique to compute plausible local averages in images (see Section 7.4.2).

8.5.1 HISTOGRAM ADJUSTMENT

Most global operators define a parametric curve with a few parameters that either may be estimated from the input image or need to be specified by the user. Histogram enhancement techniques provide a mechanism to adjust the mapping in a more fine-grained, albeit automatic, manner. Image-enhancement techniques manipulate images that are already LDR to maximize visibility or contrast. However, Ward et al. borrow key ideas from histogram enhancement techniques to reproduce HDR images on LDR displays simulating both visibility and contrast [347]. Their technique is termed "histogram adjustment."

The simulation of visibility and contrast serves two purposes. First, the subjective correspondence between the real scene and its displayed image should be preserved, so that features are only visible in the tone-mapped image if they were also visible in the original scene. Second, the subjective impression of contrast, brightness, and color should be preserved.

The histogram adjustment operator computes a histogram of a density image (i.e., the log of all pixels are taken first) to assess the distribution of pixels over all possible luminance values. The shape of its associated cumulative histogram may be directly used to map luminance values to display values. However, further restrictions are imposed on this mapping to preserve contrast based on the luminance values found in the scene and on how the HVS would perceive those values. As a

postprocessing step, models of glare, color sensitivity, and visual acuity may further simulate aspects of human vision.

The histogram is calculated by first downsampling the image to a resolution that corresponds roughly to 1° of visual angle. Then, the logarithm of the downsampled image is taken and its histogram is computed. The minimum and maximum log luminance values are taken to define the range of the histogram, except if the minimum log luminance value is smaller than -4, in which case this value is used as the lower bound of the histogram. This exception models the lower threshold of human vision. The number of bins in the histogram is 100, which provides a sufficiently accurate result in practice.

If $f(b_i)$ counts the number of pixels that lie in bin b_i, the cumulative histogram $P(b)$, normalized by the total number of pixels T, is defined as

$$P(b) = \sum_{b_i < b} f(b_i)/T$$

$$T = \sum_{b_i} f(b_i)$$

A naïve contrast equalization formula may be constructed from the cumulative histogram and the minimum and maximum display luminances:

$$\log(L_{\mathrm{d}}(x, y)) = \log(L_{\mathrm{d,min}}) + \left(\log(L_{\mathrm{d,max}}) - \log(L_{\mathrm{d,min}})\right) P(\log L_{\mathrm{w}}(x, y))$$

This approach has a major flaw in that wherever there is a peak in the histogram, contrasts may be expanded rather than compressed. Exaggeration of contrast is highly undesirable and avoidable through the following refinement.

Based on the observation that linear tone mapping produces reasonable results for images with a limited dynamic range, contrasts because of histogram adjustment should not exceed those generated by linear scaling, that is,

$$\frac{dL_{\mathrm{d}}}{dL_{\mathrm{w}}} \leq \frac{L_{\mathrm{d}}}{L_{\mathrm{w}}}$$

Because the cumulative histogram is the numerical integration of the histogram, we may view the histogram itself as the derivative of the cumulative histogram, provided it is normalized by T and the size of a bin δb:

$$\frac{dP(b)}{db} = \frac{f(b)}{T\,\delta b}$$

$$\delta b = \frac{1}{N}\log(L_{\max}) - \log(L_{\min})$$

The above naïve histogram equalization gives an expression for the display luminance L_{d} as function of world luminance L_{w}. Its derivative may, therefore, be plugged into the above inequality to yield a ceiling on $f(b)$:

$$L_{\mathrm{d}}\frac{f(\log\,(L_{\mathrm{w}})}{T\,\delta b}\;\frac{\log(L_{\mathrm{d,max}}) - \log(L_{\mathrm{d,min}})}{L_{\mathrm{w}}} = \frac{L_{\mathrm{d}}}{L_{\mathrm{w}}}$$

$$\frac{T\,\delta b}{\log(L_{\mathrm{d,max}}) - \log(L_{\mathrm{d,min}})} \geq f(b)$$

This means that as long as $f(b)$ does not exceed this ceiling, contrast will not be exaggerated. For bins with a higher pixel count, the simplest solution is to truncate $f(b)$ to the ceiling. Unfortunately, this changes the total pixel count T in the histogram, which by itself will affect the ceiling. This may be solved by an iterative scheme which stops if a certain tolerance is reached. Details of this approach are given in [347].

A second refinement is to limit the contrast according to human vision. The linear ceiling described earlier assumes that humans detect contrast equally well over the full range of visible luminances. This assumption is not correct, prompting a solution which limits the contrast ceiling according to a just-noticeable difference function δL_{t}, which takes an adaptation value L_{a} as parameter

$$\delta L_{\mathrm{t}}(L_{\mathrm{a}}) = \begin{cases} -2.86 & \text{for} \quad \log_{10}(L_{\mathrm{a}}) < -3.94 \\ (0.405\,\log_{10}(L_{\mathrm{a}}) + 1.6)^{2.18} - 2.86 & \text{for} \quad -3.94 \leq \log_{10}(L_{\mathrm{a}}) < -1.44 \\ \log_{10}(L_{\mathrm{a}}) - 0.395 & \text{for} \quad -1.44 \leq \log_{10}(L_{\mathrm{a}}) < -0.0184 \\ (0.249\,\log_{10}(L_{\mathrm{a}}) + 0.65)^{2.7} - 0.72 & \text{for} \quad -0.0184 \leq \log_{10}(L_{\mathrm{a}}) < 1.9 \\ \log_{10}(L_{\mathrm{a}}) - 1.255 & \text{for} \quad \log_{10}(L_{\mathrm{a}}) \geq 1.9 \end{cases}$$

This yields the following inequality and ceiling on $f(b)$, which also requires an iterative scheme to solve:

$$\frac{dL_\mathrm{d}}{dL_\mathrm{w}} \leq \frac{\delta L_\mathrm{t}(L_\mathrm{d})}{\delta L_\mathrm{t}(L_\mathrm{w})}$$

$$f(b) \leq \frac{\delta L_\mathrm{t}(L_\mathrm{d})}{\delta L_\mathrm{t}(L_\mathrm{w})} \cdot \frac{T \delta b L_\mathrm{w}}{\big(\log_{10}(L_{\mathrm{d,max}}) - \log_{10}(L_{\mathrm{d,min}})\big) L_\mathrm{d}}$$

The result is a practical hands-off tone-reproduction operator that produces plausible results for a wide variety of HDR images. Since the operator adapts to each image individually, the mapping of world luminance values to display values will be different for each image. As an example, two images are shown in Figure 8.57. The mappings for these two images are shown in Figure 8.58.

Further enhancements model human visual limitations such as glare, color sensitivity, and visual acuity. Veiling glare is caused by bright light sources in the periphery of vision, which cause light scatter in the ocular media. Light scatter causes a reduction of contrast near the projection of the glare source on the retina.

In dark environments, color sensitivity is lost because only one type of receptor is active. In brighter environments, the three cone types are active and their relative activity is used by the HVS to infer the spectral composition of the scene it is viewing. Finally, in dark environments, visual acuity is lost because only very few rods are present in the fovea.

The histogram adjustment technique may accommodate each of these effects, and we refer to Ward's original article for a full description [347]. Figure 8.59 shows a daytime image processed with the various different options afforded by this operator, and Figure 8.60 shows the same applied to a nighttime image.

8.5.2 BILATERAL FILTERING

The idea that an image may be separated into a high-frequency component that contains only LDR information, and a low-frequency component with an HDR, is explicitly exploited by Oppenheim's operator by attenuating low frequencies in the Fourier domain [241]. Separation of an image into separate components whereby only one of the components needs to be compressed may also be achieved by applying an edge-preserving smoothing operator.

FIGURE 8.57 *Example images "abbey," "park," "opera," and "tree" tone mapped with the histogram adjustment operator. The mappings produced for these images are plotted in Figure 8.58.*

Durand and Dorsey introduced the bilateral filter to the computer graphics community and showed how it may be used to help solve the tone-reproduction problem [74]. Bilateral filtering is an edge-preserving smoothing operator, which effectively blurs an image but keeps the sharp edges intact [322]. An example is shown in Figure 8.61, where on the right the smoothed image is shown. Edges in this image are preserved (compare with the unprocessed image on the left), while interior regions have reduced detail. This section introduces a tone-reproduction operator which uses bilateral filtering, and goes by the same name.

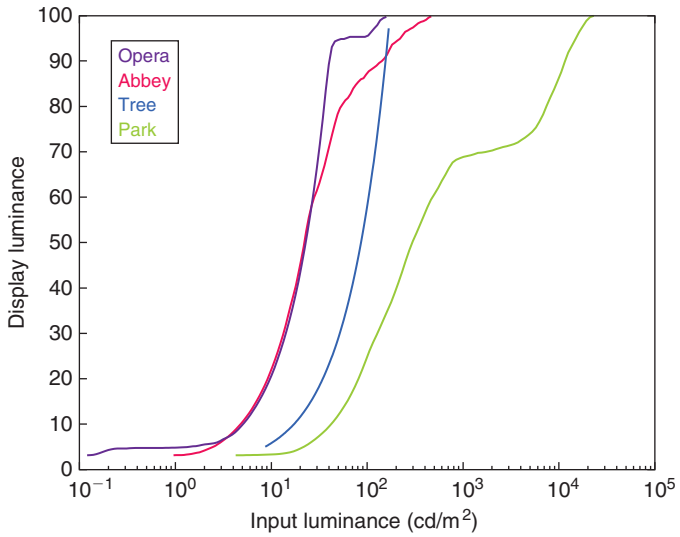


FIGURE 8.58 *Mapping of world luminances to display luminance for the images shown in Figure 8.57.*

Blurring an image is usually achieved by convolving the image with a Gaussian filter kernel. The bilateral filter extends this idea by reducing the weight of the Gaussian kernel if the density difference is too large (see Equation 8.4). A second Gaussian is applied to density differences. Following Oppenheim, this method operates on a density image, rather than on linear values.

The result of this computation, as seen in Figure 8.61, is to some extent analogous to the illuminance component as discussed by Oppenheim et al. [241]. From the input image and this illuminance image, the reflectance image may be reconstructed by dividing the input and illuminance image. The smoothed image is known as the

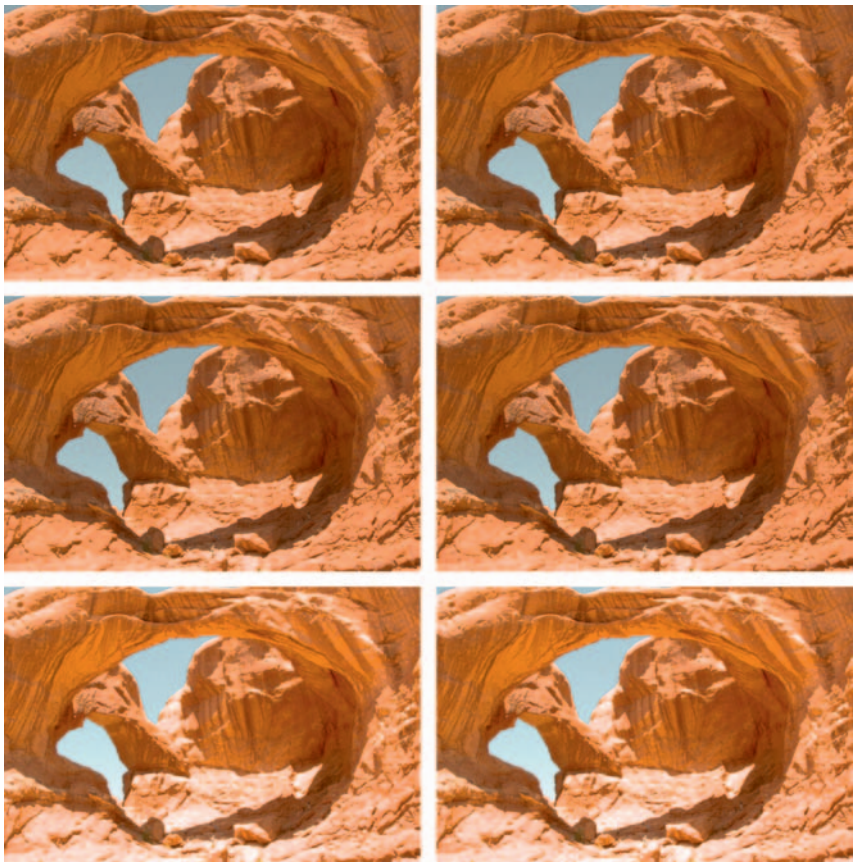


FIGURE 8.59 *Histogram adjustment with its various simulations of human visual limitations for a daylit scene. In reading order: histogram adjustment, histogram adjustment with simulation of visual acuity loss, veiling glare, color sensitivity, and contrast sensitivity, and the final image shows a combination of all of these. Compare with Figure 8.60, which shows the same techniques applied to a nighttime image.*

FIGURE 8.60 *Histogram adjustment with its various simulations of human visual limitations for a night scene. In reading order: histogram adjustment, histogram adjustment with simulation of visual acuity loss, veiling glare, color sensitivity, and contrast sensitivity, and the final image shows a combination of all of these.*

FIGURE 8.61 *The image on the left was smoothed with a bilateral filter, resulting in the image on the right.*

"base layer," whereas the result of this division is called the "detail layer." Note that the base and detail layers do not necessarily split the image into an illuminance and reflectance component. This method does not make the implicit assumption that the scene depicted is predominantly diffuse.

An example of an HDR input image, an HDR base layer, and an LDR detail layer is shown in Figure 8.62. In this figure, the bilateral filter is applied to the luminance channel only. To reconstruct the base layer in color, we replaced the luminance channel of the image (in Yxy color space) with this output and exponentiated the result to yield a linear image, and converted to RGB. The detail layer was reconstructed in a similar manner.

After the bilateral filter is applied to construct base and detail layers in the logarithmic domain, the dynamic range may be reduced by scaling the base layer to a user-specified contrast. The two layers are then recombined and the result is exponentiated and converted to RGB to produce the final displayable result.

The amount of compression that is applied to the base layer is user-specified, but Durand and Dorsey note that a target dynamic range of around 5 log units suffices

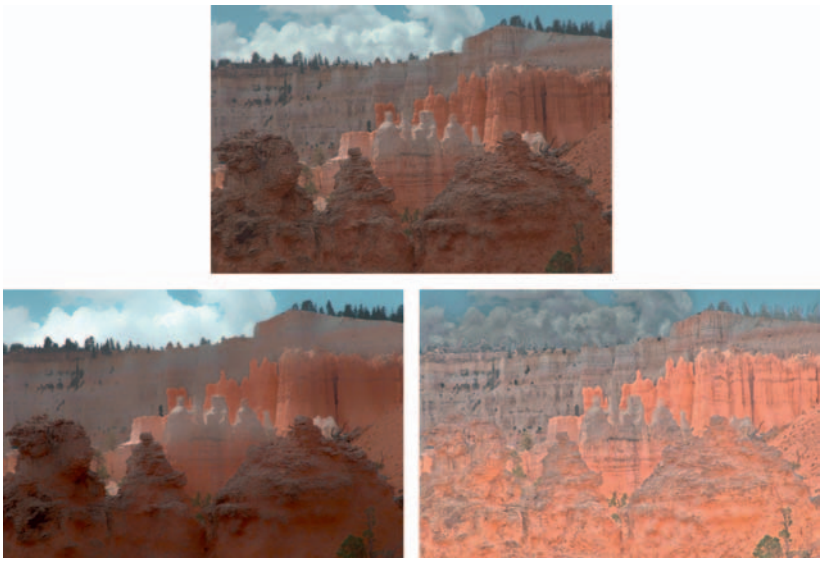


FIGURE 8.62 *HDR image tone mapped with bilateral filtering (top). The corresponding base layer and detail layers are shown in the bottom left and bottom right images.*

for many images.[4] For images that show light sources directly, this value may be adjusted. We show the effect of this parameter in Figure 8.63, where the contrast of the base layer was varied between 2 log units and 7 log units.

Bilateral filtering may be implemented directly in image space, but the convolution with a spatial Gaussian that is modulated by a Gaussian in density differences is relatively expensive to compute. In addition, the second Gaussian makes this method unsuitable for execution in the Fourier domain. Durand and Dorsey show

4 We use the natural logarithm in this case.

FIGURE 8.63 Bilateral filtering results with varying amounts of compression applied to the base layer. The dynamic range of the base layer was set between 2 log units (top-left image) and 7 log units (bottom-right image).

how these disadvantages may be overcome by splitting the density differences into a number of segments [74]. The results are then recombined afterwards, yielding an approximate solution which, in practice, is indistinguishable from accurate spatial processing. The computation is given by

$$D_j^{\text{smooth}}(x, y) = \frac{1}{k_j(x, y)} \sum_u \sum_v b_j(x, y, u, v)\ D(x - u, y - v)$$

$$k_j(x, y) = \sum_u \sum_v b_j(x, y, u, v)$$

$$b_j(x, y, u, v) = f\left(\sqrt{(x - u)^2 + (y - v)^2}\right)\ g\left(D(x - u, y - v) - D_j\right)$$

Here, the values D_j form a quantized set of possible values for pixel (x, y). The final output for this pixel is a linear combination of the output of the two smoothed values D_j^{smooth} and D_{j+1}^{smooth}. These two values are chosen such that D_j and D_{j+1} are the closest two values to the input density D of pixel (x, y).

For each segment j, the above equation may be executed in the Fourier domain, thus gaining speedup. The number of segments depends on the dynamic range of the input image, as well as the choice for the standard deviation of the Gaussian $g()$, which operates on density differences. A suitable choice for this standard deviation is around 0.4. The computation time of the bilateral filter depends on the number of segments. There is, therefore, a trade-off between computation time and visual quality, which may be chosen by specifying this standard deviation.

We have experimented with different values and show the results in Figure 8.64. For this particular image, the choice of standard deviation has a relatively small effect on its visual appearance. However, this parameter directly influences the number of segments generated, and thus affects the computation time. For this image, the largest standard deviation we chose was 8 log units, resulting in the creation of two segments. For values close to the default of 0.4, the number of segments is much higher because of the HDR of the image. This image was split into 19 segments for a standard deviation of 0.5 and into 38 segments for a standard deviation of 0.25. The computation times recorded for these images are graphed in Figure 8.65.

This computation time is substantially higher than those reported by Durand and Dorsey [74], most likely because the dynamic range of this image is higher than

FIGURE 8.64 Bilateral filtering showing results for different choices of standard deviation of the Gaussian operating on density differences, starting at 0.25 for the top-left image and doubling for each subsequent image. The bottom-right image was therefore created with a standard deviation of 8 log units.

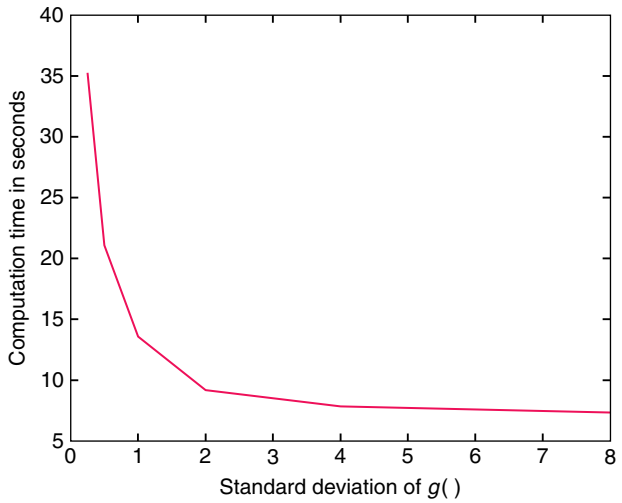


FIGURE 8.65 *Computation time of Durand and Dorsey's bilateral filter as function of the standard deviation of the Gaussian filter.*

many of their examples. In this chapter, we use a standard deviation of 0.4, as recommended in the original article, but note that discrepancies in reported computation times may be resulting from the choice of images.

Further, Durand and Dorsey observed that bilateral filtering aims at low-pass filtering, so that for most of the computations, the full resolution of the input image is not required. It is, therefore, possible to sample the image using nearest-neighbor downsampling, perform bilateral filtering, and then upsample the results to the full resolution. This significantly reduces the computational cost of the algorithm for downsampling factors of up to 10–25. Higher factors will not yield a further reduction in computation time, because upsampling and linear interpolation will then start to dominate the computation. The visual difference between no downsampling

and downsampling within this range is negligible. We, therefore, downsampled all results in this section with a factor of 16.

In summary, bilateral filtering is a worthwhile technique that achieves a hands-off approach to tone reproduction. The method is able to smooth an image without blurring across sharp edges. This makes the method robust against outliers and other anomalies. The method splits a density image into an HDR and an LDR layer. The HDR layer is then compressed and recombined with the other layer. The result is exponentiated to form an LDR image. Various techniques are available to speed up the process.

8.5.3 GRADIENT DOMAIN COMPRESSION

Fattal et al. presented a compression algorithm that achieves HDR reduction by applying a compressive function to the gradient field [89]. Following Horn [132], they compute the gradient field of a density image, manipulate these gradients, and then integrate by solving a Poisson equation.

However, rather than thresholding the gradient field, their compressive function is more sophisticated. Fattal et al. observe that any drastic change in luminance across an HDR image gives rise to luminance gradients with a large magnitude. However, fine details such as texture correspond to much smaller gradients. The proposed solution should, therefore, identify gradients at various spatial scales and attenuate their magnitudes. By making the approach progressive, that is, larger gradients are attenuated more than smaller gradients, fine details may be preserved while compressing large luminance gradients.

After computing a density image $D(x, y) = \log(L(x, y))$, the method proceeds by computing the gradient field $\nabla G(x, y)$:

$$\nabla G(x, y) = (D(x + 1, y) - D(x, y),\ D(x, y + 1) - D(x, y))$$

This gradient field is then attenuated by multiplying each gradient with a compressive function $\Phi(x, y)$ resulting in a compressed gradient field $\nabla G'(x, y)$:

$$\nabla G'(x, y) = \nabla G(x, y)\, \Phi(x, y)$$

As in Horn's approach, a compressed density image $D'(x, y)$ is constructed by solving the Poisson equation,

$$\nabla^2 D'(x, y) = \operatorname{div} G'(x, y)$$

The rationale for solving this partial differential equation is that we seek a density image $D'(x, y)$ with a gradient that approximates $G'(x, y)$ as closely as possible. In the least-squares sense, this conforms to minimizing the integral:

$$\iint \left\| \nabla D'(x, y) - G(x, y) \right\|^2 dx\, dy = \tag{8.23a}$$

$$\iint \left(\frac{\delta D'(x, y)}{\delta x} - G_{\mathrm{x}}(x, y) \right)^2 + \left(\frac{\delta D'(x, y)}{\delta y} - G_{\mathrm{y}}(x, y) \right)^2 dx\, dy \tag{8.23b}$$

According to the variational principle, $D'(x, y)$ must satisfy the Euler–Lagrange equation [258], yielding:

$$2\left(\frac{\delta^2 D'(x, y)}{\delta x^2} - \frac{\delta G_{\mathrm{x}}(x, y)}{\delta x} \right) + 2\left(\frac{\delta^2 D'(x, y)}{\delta y^2} - \frac{\delta G_{\mathrm{y}}(x, y)}{\delta y} \right) = 0$$

Rearranging terms produces the above Poisson equation, which may be solved using the full multigrid method [258]. Exponentiating the compressed density image then produces the tone-mapped image $L_{\mathrm{d}}(x, y)$:

$$L_{\mathrm{d}}(x, y) = exp(D'(x, y))$$

To a large extent, the choice of attenuation function will determine the visual quality of the result. In the previous section, a very simple example is shown by setting large gradients to zero. This produces compressed images, but at the cost of visual quality. Fattal et al. follow a different approach and only attenuate large gradients.

Their attenuation function is based on the observation that edges exist at multiple scales [369]. To detect significant ratios a multiresolution edge-detection scheme is used. Rather than attenuating a significant gradient at the resolution where it is detected, the attenuation is propagated to the full-resolution gradient field before being applied. This scheme avoids haloing artifacts.

First, a Gaussian pyramid D_0, $D_1 \ldots D_d$ is constructed from the density image. The number of levels d is chosen such that at this coarsest level, the resolution of the image is at least 32 by 32. At each level s, a gradient field $\nabla G_s(x, y)$ is computed using central differences:

$$\nabla G_s(x, y) = \left(\frac{D_s(x+1, y) - D_s(x-1, y)}{2^{s+1}}, \frac{D_s(x, y+1) - D_s(x, y-1)}{2^{s+1}} \right)$$

At each level and for each pixel, a scale factor may be computed based on the magnitude of the gradient:

$$\phi_s(x, y) = \frac{\alpha}{\|\nabla G_s(x, y)\|} \left(\frac{\|\nabla G_s(x, y)\|}{\alpha} \right)^{\beta}$$

This scale factor features two user-defined parameters α and β. Gradients larger than α are attenuated provided that $\beta < 1$, whereas smaller gradients are not attenuated and in fact may even be somewhat amplified. A reasonable value for α is 0.1 times the average gradient magnitude. Fattal et al. suggest setting user parameter β between 0.8 and 0.9, although we have found that larger values up to around 0.96 are sometimes required to produce a reasonable image.

The attenuation function $\Phi(x, y)$ can now be constructed by considering the coarsest level first and then propagating partial values in top-down fashion:

$$\Phi_d(x, y) = \phi_d(x, y)$$
$$\Phi_s(x, y) = U(\Phi_{s+1}(x, y))\, \phi_s(x, y)$$
$$\Phi(x, y) = \Phi_0(x, y)$$

Here, $\Phi_s(x, y)$ is the partially accumulated scale factor at level s, and $U()$ is an upsampling operator with linear interpolation.

For one image, the two parameters α and β were varied to create the tableau of images shown in Figure 8.66. For smaller values of β, more details are visible in the tone-mapped image. A similar effect occurs for decreasing values of α. Both parameters afford a trade-off between the amount of compression that is applied to the image and the amount of detail that is visible. In our opinion, choosing values that are too small for either α or β produces images that contain too much detail to appear natural.

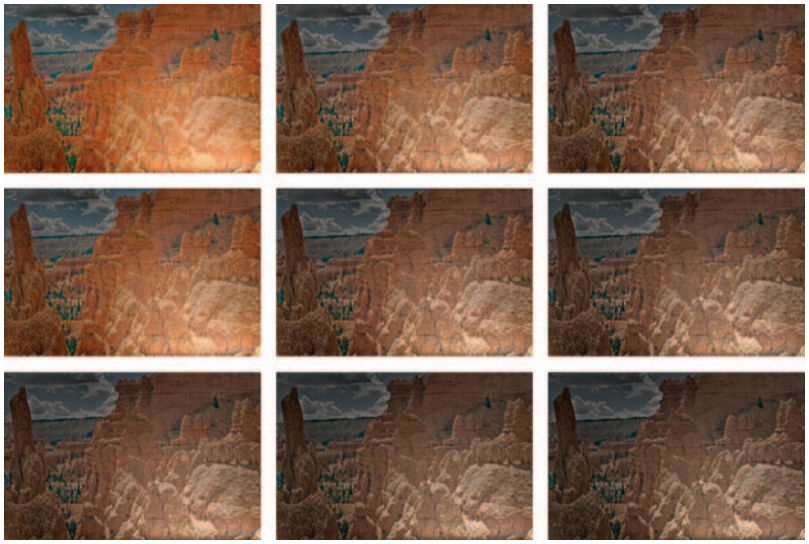


FIGURE 8.66 *Fattal's gradient domain compression. The user parameters α and β were varied: from left to right, α is given values of 0.10, 0.25, and 0.40. From top to bottom, β is 0.85, 0.89, and 0.95.*

This approach may benefit somewhat from clamping, a technique whereby a percentage of the smallest and largest pixel intensities is removed, and the remaining range of intensities is scaled to fit the display range. In Figure 8.67, we show the effect of varying the percentage of dark pixels that are clamped (top row) and separately the percentage of light pixels that are clamped (bottom row). The effect of clamping dark pixels is fairly subtle, but dramatic effects may be achieved by clamping a percentage of light pixels. In general, if a tone-mapped image appears too gray, it may be helpful to apply some clamping at the light end. This removes outliers that would cause the average luminance to drop too much after normalization.

In summary, Fattal's gradient domain compression technique attenuates gradients, but does so in a more gentle manner than simply thresholding. The two user parameters provide a trade-off between the amount of compression and the amount of detail available in the image. Too much compression has the visual

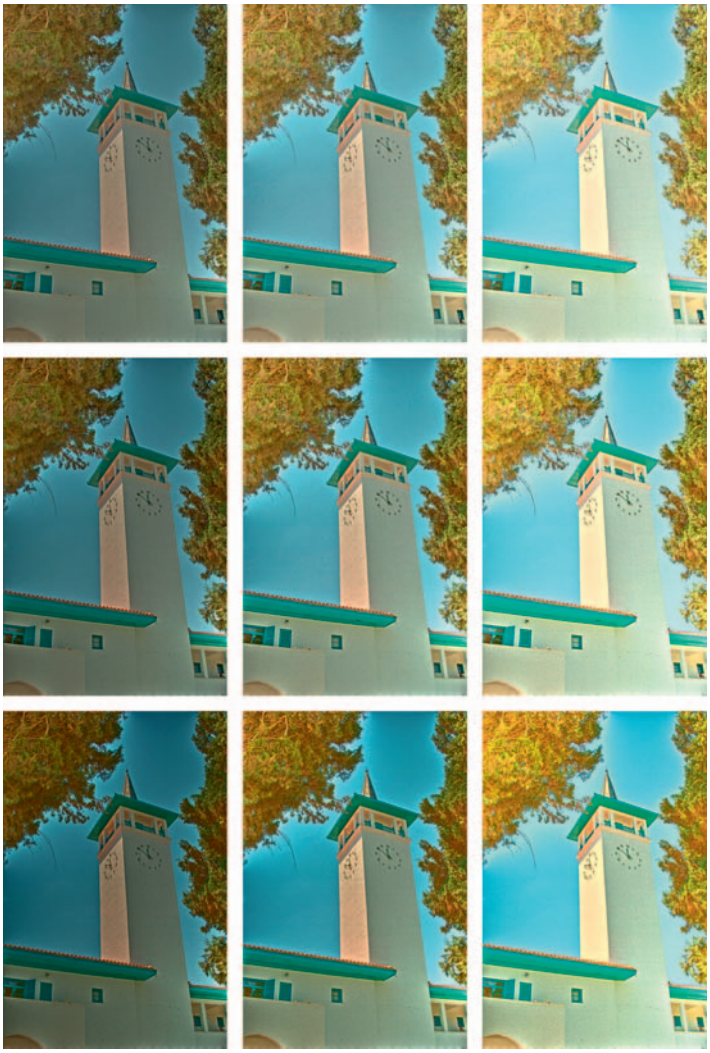


FIGURE 8.67 Clamping in Fattal's gradient domain compression. Left to right: clamping 0.1%, 1%, and 10% of the light pixels. Top to bottom: clamping 0.1%, 1%, and 10% of the dark pixels.

effect of exaggerated small details. The technique is similar in spirit to Horn's lightness computations and is the only recent example of a tone-reproduction operator working on gradient fields.

8.5.4 MULTISCALE OPTIMIZATION FRAMEWORKS

Gradient domain tone mapping as presented in the previous section deals only with contrasts between neighboring pixels. As can be seen in Equation 8.23, which is the cost function to be minimized in the least-squares sense, the constraints are imposed only on local contrast, while global contrast may change freely. A minimization formulated in this manner may lead to visible distortions of low spatial frequencies in the reconstructed image, as the local enhancement of high spatial frequencies may not leave enough dynamic range to fully accommodate global contrast relations from the original image. Visually, this may lead to excessively "flattened" global contrast and unnaturally enhanced image details.

A simple way to account for contrast between more distant image regions is to consider all pairs of pixels in the minimization formulation, which in practice may be too costly for high-resolution images [105]. Mantiuk et al. [208] have proposed a trade-off solution where they consider larger pixel neighborhoods for local contrast and also several levels of a Gaussian pyramid for global contrast. This way, the control over local and global contrast still relies on the optimization procedure. Similar to Fattal's et al. approach, this produces halo-free and extremely sharp images. Figure 8.68 shows a comparison between the two techniques.

Note that Mantiuk et al. represent contrast as a low-pass (Gaussian) pyramid in logarithmic space [250]. This means that local contrast is stored at the finest pixel-level scale, whereas global contrast is stored more sparsely for larger image patches (i.e., squared pixel blocks) at higher pyramid levels. This is in agreement with the photoreceptor distribution in the retina, where through foveal vision (spanning only about 1.7° of visual angle) the differences between fine image details can be readily judged. The same is impossible for similar differences between single pixels that are spread apart in the field of view. In the latter case, parafoveal vision is involved (spanning 160° of visual angle), which has almost no ability to process high-frequency information, so that distant pixel patches can effectively be compared only if they are large enough. This justifies using a more sparse global contrast representation, which accounts for contrasts between large image patches rather than single distant pixels.

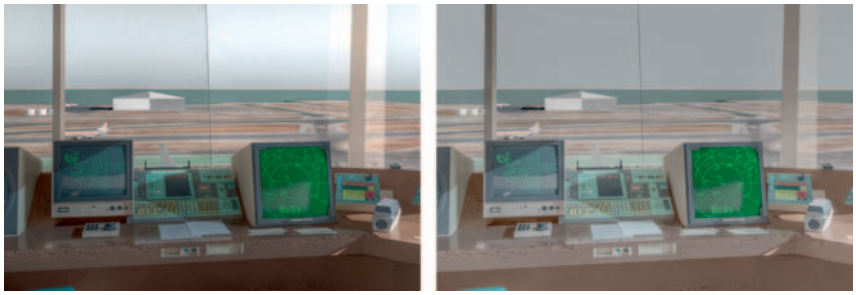


FIGURE 8.68 *The algorithm by Fattal et al. (left) does not take into account global contrast relations, leading to different brightness for the window panes in the areas where the original luminance is the same. Proper global contrast relations can be restored at the expense of local contrast loss, when a multiscale contrast representation is used for the HDR contrast compression (right). (Images courtesy of Rafał Mantiuk)*

Another perception-inspired step in the multiscale contrast processing as proposed by Mantiuk et al. is the transduction of logarithmic contrast (i.e., physical contrast) into a hypothetical response to such contrast by the HVS (i.e., perceived contrast). Mantiuk's transducer (Section 10.9.2) is used for this purpose, which is a simple monotonic function, described by Equation 10.31. It takes into account a nonlinearity in the contrast discrimination thresholds for different pedestal (background) contrasts because of contrast self-masking (see Sections 10.7.5 and 10.9). This way, the transducer models the lower sensitivity of the eye to contrasts below the detection threshold, so that these contrasts remain invisible. It also models high-contrast ranges, in which case either neural noise or contrast inhibition limits our vision (see also Figure 10.22 for a plot of a typical contrast transducer).

Such differences in contrast sensitivity have been used by Mantiuk et al. in multiscale gradient domain tone mapping where stronger contrasts can be compressed more without introducing visible differences in the image. However, contrast compression should be minimized in the middle contrast range, where the eye is the most sensitive. The tone-mapping operator that follows such guidelines for multiscale contrast representation is called "contrast mapping" [208].

Another approach, called "contrast equalization," which leads to even sharper but less natural-looking images, performs histogram equalization on the same multiscale contrast representation. The motivation for equalizing the histogram of contrasts is to allocate dynamic range for each contrast level relative to the space it occupies in an image. The perceived contrast values resulting from both contrast compression approaches are then converted back to physical logarithmic contrasts using the inverse transducer (Equation 10.34), and then finally to luminance values by solving an optimization problem over all contrast scales. Note that for the histogram equalization approach, the basic difference with respect to the so-called histogram adjustment operator (described in Section 8.5.1) is that the latter one operates directly on luminance values while Mantiuk et al. consider contrast values instead.

Contrast domain processing offers several advantages over operating directly in the luminance domain. First, contrast can be modified with regard to its magnitude, thus it becomes simple to take advantage of contrast perception characteristics such as contrast detection and discrimination thresholds, which are embedded in the transducer function. Second, extremely sharp contrasts can be achieved without introducing halo artifacts (refer to Figure 8.69) because contrast values are compressed but not reversed. Finally, the pyramid of contrast values, which is in good agreement with foveal and parafoveal contrast perception, can take into account both local (high-spatial-frequency) and global (low-frequency) contrast relations.

This low-pass multiscale contrast measure has the advantage that by modifying contrast selectively at any scale, halo artifacts can be avoided. However, the explicit manipulation of selected spatial frequencies (band-pass selectivity) is limited. Such explicit control can be achieved by band-pass multiscale contrast definitions [250], although in these approaches any stronger per-band contrast manipulation inevitably leads to halo artifacts (see Section 8.2.1).

Farbman et al. [86] addressed this problem by introducing an edge-preserving operator based on the weighted least squares (WLS) framework, which results in a very elegant multiscale image decomposition. The basic idea behind this approach is that for each edge signal, its complete frequency domain representation is kept only at one scale. This way, each enhancement performed over a given scale leads to enhancing the edge signal fully contained in this scale, which naturally prevents any halo artifacts.

The choice of scale to which a given edge belongs depends directly on its contrast, where larger contrast edges are associated with lower spatial frequency scales. This follows the intuition that image features of wider spatial extent should have higher contrast, which is usually true in real-world scenes (see also [312] for a discussion of this issue and interesting counterexamples). The multiscale image decomposition

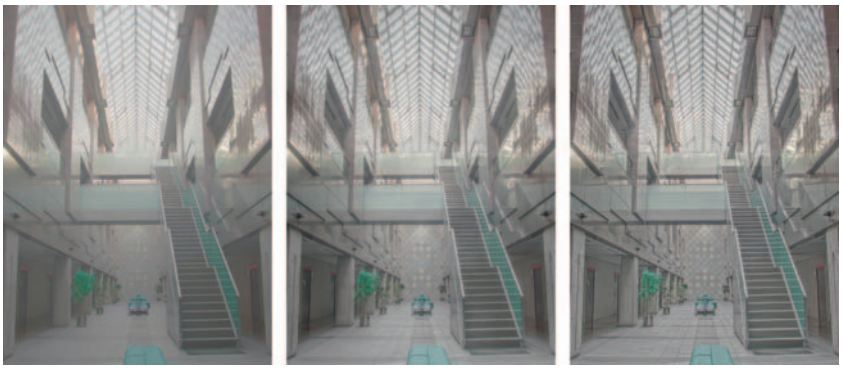


FIGURE 8.69 *The linear rescaling of luminance in the logarithmic domain (left) compared with the two discussed contrast compression methods: contrast mapping (middle) and contrast equalization (right). (Images courtesy of Rafał Mantiuk; source HDR photograph courtesy of Frédéric Drago)*

is achieved by iterative application of an edge-stopping image-smoothing operator, which is tuned in subsequent iterations to preserve only edges of larger and larger contrast (refer to the top row in Figure 8.70). The operator is designed as an optimization procedure, which penalizes for image gradients (smoothing effect) in the whole image except near strong edges (edge-stopping effect).

Although the WLS framework is designed as a general tool for image enhancement and manipulation, it can also be used for multiscale tone manipulation. In this case, the WLS-based smoothing is applied to separate base and detail layers as in bilateral filtering (Section 8.5.2), where contrast in the base layer is compressed. The main advantage of WLS-based decomposition is avoiding mild halo artifacts, which might sometimes be visible when bilateral filtering is used for such decomposition.

Farbman et al. also experiment with decomposing HDR images into four bands and then applying different scaling factors to each of them. They observed that by strong compression of the base layer and by boosting the finest scale layer, they obtain an appearance similar to gradient domain operators, demonstrated in the bottom-middle image of Figure 8.70. However, by using less compression of the base layer and moderate boosting of coarser detail layers, an image appearance simi-

lar to the photographic tone reproduction (Section 8.1.2) can be obtained, as shown in the bottom-right image of Figure 8.70.

GPU implementations of Mantiuk et al. and Farbman et al.'s techniques enable multiscale image decompositions at interactive speeds. Even better performance can be achieved using another multiscale representation based on edge-avoiding lifting scheme wavelets as proposed by Fattal [88]. The wavelet functions are image-content-adaptive, and their support never crosses high-contrast edges. This enables band-independent multiscale processing, similar to the WLS approach, as the correlation between bands at strong edges is minimized. The main advantage of the edge-avoiding wavelets is their computational efficiency.

8.6 EXPOSURE FUSION

Two or more images of the same scene may be combined into a single image, which is in some way more informative, more suitable for visual perception, or better for image processing [110]. Such techniques are collectively known as "image fusion" and serve to reduce uncertainty and maximize the amount of relevant information in the resulting image, possibly specific to a particular task. An example is the way in which HDR images are created by means of combining multiple exposures. These algorithms are discussed in Chapter 5.

In this chapter and in the preceding chapter, we have shown many tone-reproduction operators that create 8-bit output. These images are frequently better, in some sense, than any of the original exposures from which they are created. This insight has led to the development of algorithms that combined multiple exposures directly into an 8-bit LDR image. Thus, multiexposure techniques discussed in Chapter 5 combine exposures into HDR images; we briefly discuss here a collection of algorithms known as "exposure fusion" [217]. They incorporate the notion of combining multiple exposures with tone reproduction.

There are both advantages and disadvantages to exposure fusion. Skipping the HDR stage simplifies the acquisition pipeline. Algorithms can be engineered such that camera response curves do not need to be recovered, nor is it necessary to know the precise exposure time for each of the input images. Moreover, it is possible to include flash images in the sequence [217].

However, by skipping the construction of an HDR image, its advantages are also lost. In particular, it will be more difficult to do certain types of image processing such as white balancing and simulating motion blur. Moreover, the resulting images cannot be used as environments in image-based lighting algorithms (see Chapter 11).

FIGURE 8.70 Tone mapping using the WLS multiscale method. An HDR image decomposed into four scales: a baseband and three detail layers is considered. The top row depicts the outcome of coarsening of the image because of iterative subtracting of detail layers from fine to coarse scales (from left to right). Note that edges with sufficiently strong contrast are preserved across subsequent coarsening iterations, which eventually results in piecewise constant image regions separated by strong edges. The bottom row shows tone-mapping results: (left) the linear rescaling of luminance in the logarithmic domain, (middle) the WLS contrast compression for the baseband (50%) and enhancement for the fine detail layer (200%), (right) the WLS contrast compression for the baseband (60%) and enhancement for the low and medium detail layers (170%). (Images courtesy of Dawid Pająk; source HDR photograph courtesy of Piotr Didyk)

Perhaps the most serious disadvantage of exposure fusion is that the resulting image is prepared for a specific display and viewing environment, typically a hypothetical average display. If better display technology comes along, such as HDR display devices, the fused image cannot be adjusted to take advantage of higher dynamic ranges. Also, an average viewing environment is assumed, so that viewing an image under different conditions will be suboptimal.

Although several algorithms for exposure fusion exist [109,263], we focus on one specific algorithm as a good example of this type of technique, which is the one proposed by Mertens et al. [217]. This algorithm assumes that all exposures are in perfect registration. The idea is then to keep the best parts of each exposure, which is guided by quality measures that produce a weight map for each exposure. The fused image is then generated by blending the exposures according to the weight maps.

The weight maps are computed on the basis of three quality criteria. First, by applying a Laplacian filter and recording the absolute value of the filter response, a measure of contrast is obtained. High values indicate edges, which are then given a high weight in the weight maps. Second, on the basis that saturation is desirable, but tends to be less present in longer exposures, higher weights are assigned to saturated colors. This is computed at each pixel as the standard deviation within the red, green, and blue channels. Finally, a measure of well-exposedness is included. This is akin to the weight function applied in multiexposure techniques (see Figure 5.3). However, here a Gaussian function is used to measure how close a normalized pixel value is to 0.5. The standard deviation used is $\sigma = 0.2$. The weight map for each exposure k and for each pixel (i, j) is constructed by multiplying the weights for these three measures, leading to a final weight $w_{i,j,k}$.

The fusion algorithm begins by normalizing the weights for all N exposures:

$$\hat{w}_{i,j,k} = w_{i,j,k} \Big/ \sum_{n=1}^{N} w_{i,j,n} \tag{8.24}$$

Assuming that the recorded pixel values (i, j) in exposure k are indicated with $Z_{i,j,k}$, the resulting image I could be obtained by summing over all pixels, incorporating the weight values:

$$I_{i,j} = \sum_{k=1}^{N} w_{i,j,k} \, Z_{i,j,k} \tag{8.25}$$

Unfortunately, this would not lead to satisfactory results, creating undesirable seams in places where the weights vary with high spatial frequencies. This is an artifact of the different absolute pixel values found in different exposures for the same regions in the scene.

A reasonable solution is afforded by constructing a Laplacian pyramid and combining pixels at each level in the pyramid separately. With the Laplacian pyramid of image Z at level m given by $L(Z_k)^m$ and the weight map w decomposed into a Gaussian pyramid $G(w_k)^m$, the new Laplacian pyramid $L(I)^m$ can be constructed as follows:

$$L(I)^m_{i,j} = \sum_{k=1}^{N} G(w_{i,j,k})^m \; L(Z_{i,j,k})^m \tag{8.26}$$

FIGURE 8.71 *Exposure fusion results using the technique of Mertens et al. [217]. The individual exposures are shown on the left, whereas the final result is shown on the right.*

FIGURE 8.72 Further exposure fusion results using the technique of Mertens et al. [217].

The resulting Laplacian pyramid $L(I)^m$ is then collapsed to generate the final image I.

Such multiresolution approaches are effective at avoiding seams. An example generated with this method as well as its constituent exposures are shown in Figure 8.71. Two further example results are shown in Figure 8.72. Note that we have chosen images which are well-aligned. If the exposures are not well-aligned, the individual exposures could first be aligned, perhaps with the alignment algorithm discussed in Section 5.4.

Inverse Tone Reproduction

09

The capabilities of some advanced display systems are currently exceeding the dynamic range of conventional 8-bit images by a significant amount. Such high dynamic range (HDR) displays are discussed in Chapter 6. We anticipate that such displays will become more numerous and widespread in the near future. A concern for the general acceptance of such displays is the ability to generate appropriate content to drive such displays.

Of course, there exist both hardware and software techniques to capture HDR images and video directly, as outlined in Chapter 5. However, given that a wealth of images and videos are currently available as 8-bit low dynamic range (LDR) data, it is an interesting proposition to consider the possibility of upscaling conventional images for display on HDR display devices. The general collection of techniques that takes as input an LDR image to produce an HDR representation is referred to as "inverse tone reproduction."

While in tone reproduction the dynamic range is reduced, here the dynamic range is expanded. To expand an image, several issues have to be taken into account. First, the shape of the curve that accomplishes the dynamic range expansion needs to be determined. Second, there may be under- and overexposed areas in the image, which would have to be treated with some spatial processing to fill in the missing information. Finally, expansion of quantized signals may lead to visible contouring (banding) artifacts. In particular, this can be a problem for lossy compression techniques. Note that JPEG compression works by allowing the quantized image data to be replaced with block encodings that approximate the original data. The

© 2010, Elsevier Inc. All rights reserved.
DOI: 10.1016/B978-0-12-374914-7.00009-9

encoding is carried out in such a way that the error of the approximation is kept below the visible threshold. However, if we expand a JPEG-encoded image, we may inadvertently expand the luminance values in such a way that the block structure becomes visible.

In this chapter, we discuss all three issues. The following section discusses the various expansion functions that have been proposed. Section 9.2 discusses a technique that could be used to salvage under- and overexposed material. Section 9.3 outlines some techniques for reducing the visibility of quantization and lossy encoding artifacts. Finally, in Section 9.4, we discuss some user studies that have recently appeared and that assess which type of expansive function leads to material that is most preferred by viewers. Such studies provide important insights as to how inverse tone-reproduction operators can be designed that maximize the viewer experience.

9.1 EXPANSION FUNCTIONS

To prepare an LDR image for display on a device with a higher dynamic range than encoded in the image, we usually begin by expanding the range of values in the image. In essence, each pixel's luminance value is passed through a function that returns new luminance values that are larger than or equal to the input values. Generally, these functions are monotonically increasing so that the introduction of haloing artifacts can be avoided.

9.1.1 EXPANDING HIGHLIGHTS

To guide the shape of such dynamic range expansion functions, additional criteria can be specified. For instance, one could argue that highlights and light sources in images need to be expanded more than pixels depicting less specular or nonemissive surfaces [220]. Under these assumptions, it would be possible to analyze image histograms for the purpose of detecting highlights and light sources and expanding these pixels more than the remainder of the image.

Figure 9.1 illustrates a simple inverse tone reproduction based on this principle [220], where the scaling function s_1 is used to expand all pixels in the image except highlights and light sources, which are even more aggressively expanded by the steeper scaling function s_2. The splitting point between the two scaling functions is

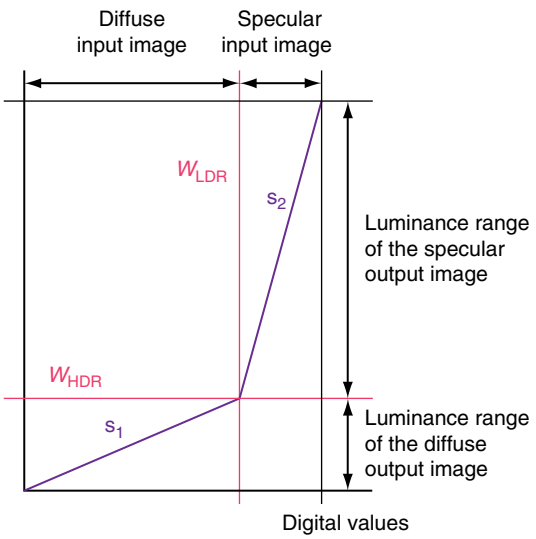


FIGURE 9.1 *Display dynamic range splitting between diffuse image regions and highlights/self-emissive objects. W_{LDR} denotes the maximum diffuse white in the input LDR image, and W_{HDR} denotes the corresponding intensity in the enhanced HDR image. (Image courtesy of Laurence Meylan of General Electric, Scott Daly of Sharp Laboratories of America, and Sabine Süsstrunk of Ecole Polytechnique Fédérale de Lausanne. Redrawn from [221])*

decided based on the maximum diffuse white W_{LDR} in the input LDR image (refer to [162] for a discussion of reference white extraction).

An interesting question is how to select W_{HDR}, which splits the dynamic range in the reconstructed HDR image, so that its preferred appearance on an HDR display can be obtained. Meylan et al. [220] performed a psychophysical experiment to investigate this issue. It turns out that for indoor scenes, more preferred results are obtained when more dynamic range is allocated for highlights.

Conversely, for outdoor scenes, an overall brighter image appearance is preferred, which leaves less dynamic range to specular highlights. Further the relative size of highlights might be important, where for larger highlights, overall dimmer images are preferred (e.g., the sun reflecting in the water surface, which should remain dim). However, the general trend is that boosting brightness of specular highlights leads to a more natural appearance than a uniform expansion of dynamic range for all pixels (see Figure 9.2).

In fact, pure histogram analysis of the input LDR image might not always be reliable in detecting highlights and in deriving W_{LDR}. Meylan et al. [221] observe that due to sharp changes of illumination near highlights, these regions usually contain significantly more high spatial frequencies than diffuse regions. It can be shown that by using a set of low-pass filters combined with morphological operations, highlights can be robustly extracted. Figure 9.2 (bottom) displays an example where such automatically detected highlights are marked in red.

Note that the main aim here is to boost the brightness of automatically detected highlights and self-luminous objects. This is often sufficient to obtain a plausible appearance of LDR images on HDR displays. One may expect that in an HDR image, the actual pattern of texture in the highlight region and filaments of luminaires should be reproduced. We address the problem of recovering image content in under- and overexposed regions in Section 9.2.

9.1.2 INVERSE PHOTOGRAPHIC OPERATOR

Alternatively, it is possible to simply take an existing tone-reproduction operator and invert it. Most global tone-reproduction operators define simple monotonic curves and are therefore amenable to inversion. Banterle et al. [18] experiment with a number of such simple tone-mapping operators and found that the most visually compelling results have been obtained by inverting the photographic tone-mapping operator [274]. Using the same notation as in Section 8.1.2, the desired expanded values $L_{\mathrm{w}}(x, y)$ are related to the LDR input $L_{\mathrm{d}}(x, y)$ according to

$$\frac{\alpha^2}{L_{\mathrm{white}}^2 \bar{L}_{\mathrm{w}}^2} L_{\mathrm{w}}^2(x, y) + \frac{\alpha}{\bar{L}_{\mathrm{w}}} \left(1 - L_{\mathrm{d}}(x, y)\right) L_{\mathrm{w}}(x, y) - L_{\mathrm{d}}(x, y) = 0 \qquad (9.1)$$

FIGURE 9.2 Dynamic range enhancement: (left) linear scaling, (right) independent linear scaling for diffuse and strongly specular/emissive objects as proposed in [220], and (bottom) automatic detection of highlights and self-luminous objects (all marked in red) in LDR images using a segmentation algorithm as proposed in [221]. Note that an HDR display is required to reproduce the appearance of highlights in the right image, which has been rendered for $W_{\mathrm{HDR}} = 67\%$. The bottom image shows the original appearance as intended for an LDR display. In this respect, linear scaling (left) leads to an excessively bright image, while independent highlight scaling (right) results in a more faithful reproduction of appearance. (Images courtesy of Laurence Meylan of General Electric, Scott Daly of Sharp Laboratories of America, and Sabine Süsstrunk of Ecole Polytechnique Fédérale de Lausanne; reprinted from [221], page 649210-9 with permission from SPIE)

Here, α is the key value that is input by the user, L_{white} is the smallest luminance value that is mapped to white, and $\bar{L}_{\text{w}}$ is the geometric average of the image.

Banterle et al. [18] observe that when using the inverted photographic tone-mapping operator, the amount of expansion is limited due to quantization errors (contouring artifacts), which become visible in particular in bright image regions. This is clearly related to the sigmoidal shape of the tone-mapping curve that features strong contrast compression in dark and bright image regions. Due to the inversion of the sigmoid, contrast in such regions becomes excessively expanded given a small number of bits used for its encoding in the source LDR image. The issue of reducing the visibility of such contouring (banding) artifacts will be discussed in Section 9.3.

So far, the main objective of using inverted tone mapping is to obtain a visually plausible HDR image. However, in many applications, it is desirable that HDR pixels represent the actual radiance values in the scene. This requires recovering the camera response function, which captures the relation between the irradiance at the camera sensor and the pixel intensity code values in the image. This means that in principle, the scene radiance map can be reconstructed by inverting the camera response function. In fact calibration to absolute radiance units might also be needed, which is equivalent to simple rescaling of the whole map based on the radiance measurement for at least 1 pixel.

In practice, in modern cameras, intensive image processing is performed by camera firmware. This includes noise reduction, color interpolation from the Bayer pattern, and contrast and color saturation enhancement. Each of these processing steps may affect the accuracy of the camera response derivation. In Section 5.7, it is shown how the camera response can be recovered using a set of differently exposed images of the same mostly static scene. However, often only a single legacy image is available, and no assumption can be made about the scene characteristic, camera type, or its exposure parameters. The problem then arises of how to reconstruct the camera response based on such a single image.

9.1.3 GAMMA EXPANSION

In many practical applications, a simple gamma correction curve (e.g., with a power coefficient value of 2.2) is assumed as a sufficiently good approximation of the camera response function [335]. The so-called blind inverse gamma correction as proposed by Farid [87] leads to a more precise estimate of the actual gamma

correction in the image. Frequency analysis reveals that gamma correction introduces new spatial frequencies with amplitude and phase correlated to the original spatial frequency content in the image. Such correlations monotonically increase with increasing nonlinearity of the gamma correction and can be detected using tools from polyspectral analysis. By applying inverse gamma, which minimizes such correlations, the actual gamma correction originally applied to the image can be found.

Lin et al. [188] show that the actual camera response can be approximated more accurately by analyzing the distribution of color pixels across strong edges in the image. In this respect, the most meaningful edges are those that separate uniform scene regions with sufficiently different colors and radiance values R_1 and R_2 as shown in Figure 9.3(a). Figures 9.3(b) and 9.3(c) show the corresponding sensor

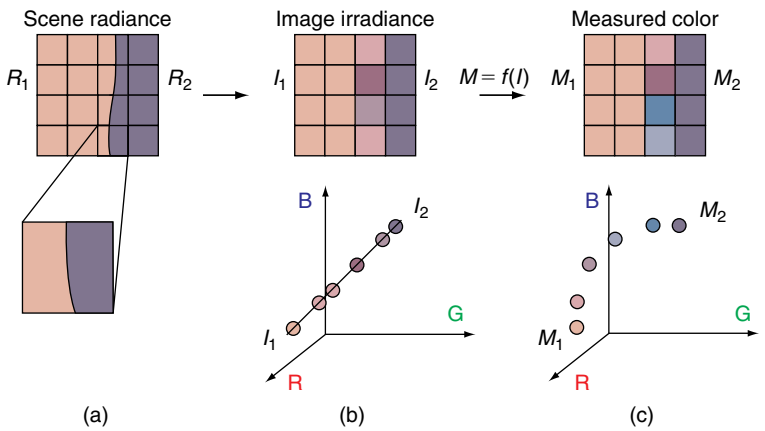


FIGURE 9.3 *Nonlinear distribution of measured colors in the edge region. (a) Two scene regions separated by an edge feature distinct radiance values R_1 and R_2, which in (b) is projected at the camera sensor pixels as irradiance values I_1 and I_2. Colors of linearly interpolated pixels at the sensor lie on the line between I_1 and I_2. (c) A nonlinear camera response $M = f(I)$ warps these colors, which results in their nonlinear distribution (Redrawn from [188])*

irradiance I_1 and I_2 and measured image color $M_1 = f(I_1)$ and $M_2 = f(I_2)$, where $M = f(I)$ denotes the camera response. For pixels located at the edge, the irradiance value I_p can be expressed as a linear combination $I_p = \alpha I_1 + (1-\alpha)I_2$, where α depends on the partial coverage of each of the two regions by the edge pixel. Then, for a camera with a linear sensor response, the corresponding measured color M_p can be expressed as

$$M_p = \alpha M_1 + (1-\alpha)M_2 = f[\alpha I_1 + (1-\alpha)I_2] \tag{9.2}$$

However, for a camera with a nonlinear response, the actual measured color M_p at the edge may be significantly different from such a linear combination (see the bottom plot in Figure 9.3[c]). By identifying a number of such edge color triples $<M_1, M_2, M_P>$ and based on the prior knowledge of typical real-world camera responses, a Bayesian framework can be used to estimate the camera response function. Lin et al. report a good accuracy of radiance maps obtained using this method, where the best performance can be expected when edge color triples $<M_1, M_2, M_P>$ cover a broad range of radiances for each color channel.

The method's accuracy can be affected when only a limited range of colors is available in the image. On the other hand, by considering edge color triples from other images captured with the same camera, the accuracy of the camera response estimation can be significantly improved.

9.1.4 LDR2HDR

An alternative approach for contrast boosting was presented by Rempel et al. [275]. The focus of this approach is on on-the-fly processing of arbitrary video sources without any human intervention. The ultimate goal of this approach is to be implemented directly in HDR display devices, where it can be used to improve the image quality of LDR video sources, such as broadcast television programs or DVD/Blueray content.

As such, the focus of the method is significantly different from some of the other works discussed in this chapter; unlike methods such as the one by Banterle et al. [18], Rempel et al. anticipate that their method would be applied to original LDR footage that has not previously undergone a tone-mapping step.

Combined with the need for a completely automatic method, this mandates a conservative approach, whose primary concern is not to create any artifacts.

The method consists of two primary components, as depicted in Figure 9.4. First, an inverse gamma stage maps pixel values into linear luminance in order to compensate for the nonlinear encoding of intensities in LDR images. Those luminance values are linearly stretched to a contrast of about 5000:1, similarly to the work by Akyüz et al. [6]. An image filter is applied in order to suppress noise, as well as quantization and image compression artifacts.

The second component is the computation of a *brightness enhancement* function, which is multiplied into the image to increase brightness in image regions containing saturated pixels, that is, pixels with at least one channel near the maximum value. A smooth version of this function is computed by first determining a binary mask of pixels where at least one color channel exceeds a certain threshold intensity. This mask is blurred with a large-kernel Gaussian filter whose support is designed

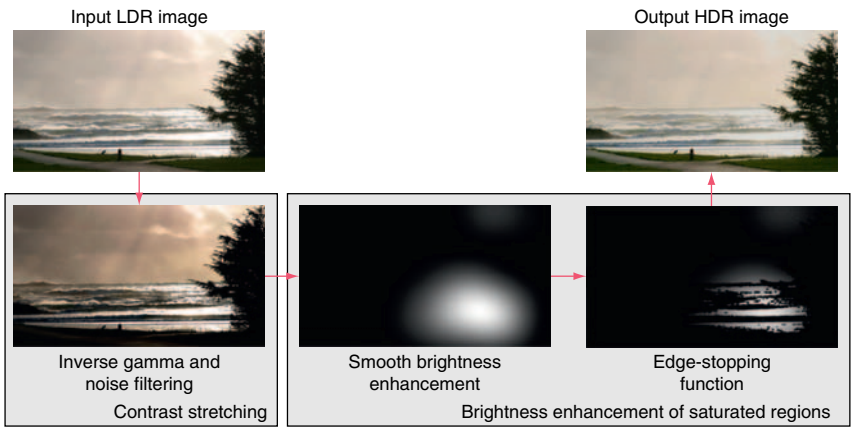


FIGURE 9.4 *Schematic overview of the LDR2HDR method by Rempel et al. [275].*

such that it creates spatial frequencies that lie in a frequency band to which the human visual system is not very sensitive [237,246]. By designing the brightness enhancement function in this way, one can increase the intensity in the neighborhood of saturated image regions without introducing visible artifacts. The authors propose to use a Gaussian with a standard deviation corresponding to roughly 1.2° viewing angle, as seen from the intended viewing distance.

Before the brightness enhancement function is applied to the image, its impact is limited to the bright side of regions with strong edges. This is achieved with the help of an edge-stopping function that is also computed from the initial mask through a flood-fill operation, which uses strong edges as boundaries. The product of the smooth function and the edge-stopping function forms the final brightness-enhancement function.

One challenge with this basic algorithm is that the flood-fill operation and the large-radius Gaussian filter provide challenges for real-time implementations, especially on embedded platforms in consumer devices. The authors, therefore, also describe a hierarchical implementation based on image pyramids, which is much more efficient. Figure 9.5 illustrates this hierarchical version of the algorithm. The large Gaussian blur is implemented by successively downsampling the mask representing the saturated pixel regions (1) and then upsampling it again with nearest-neighbor interpolation, while applying a small (3×3 pixel) Gaussian blur at each level (2). To compute the edge-stopping function, edge image is first generated at the highest resolution as described above and then downsampled into an image pyramid (3). The actual edge-stopping function is then created through a sequence of upsampling and dilation operations from the lowest resolution binary saturation mask (4). As with the smooth brightness enhancement function, the upsampling for the edge-stopping function uses nearest-neighbor interpolation. The dilation operators use a 3×3 binary mask (4) but stop at pixels that are marked as an edge in the edge image of the corresponding resolution (3).

Note that the radius of the dilation at each level (4) is the same as that of the blur on the corresponding level of the upsampling pyramid (2) so that the blur and the edge-stopping function propagate outward at the same speed. Note also that the edge-stopping function can have hard edges in smooth image regions. However, these edges are outside the area of influence of the blur function and thus do not create discontinuities in the final image.

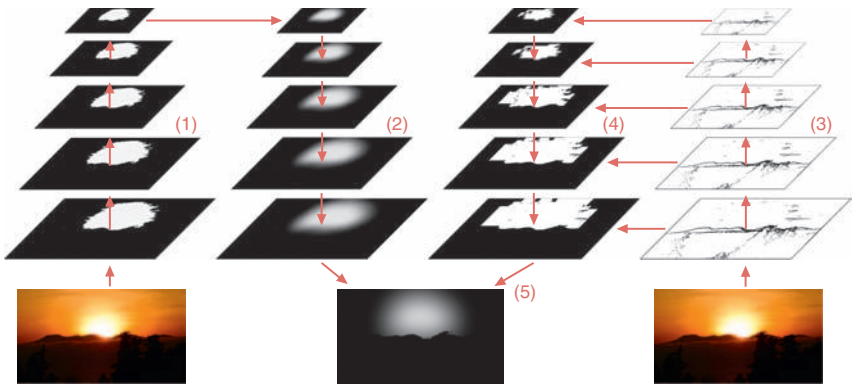


FIGURE 9.5 *The hierarchical version of the LDR2HDR algorithm.*

9.2 UNDER- AND OVEREXPOSED MATERIAL

For all methods discussed in the preceding section, the HDR image content cannot be reconstructed for clipped dark and bright image pixels. We address this issue here.

The problem of lost information reconstruction in under- and overexposed image regions is clearly underconstrained with a number of possible solutions that lead to the same appearance of an LDR image. The most promising results are obtained so far using inpainting and texture synthesis techniques specialized in repairing damaged images or removing unwanted objects. Since clipped image regions may still contain sparse information, learning approaches that rely on finding correspondences in a predefined database of LDR and HDR image pairs is also an interesting avenue of research. In this section, we present successful examples of both approaches, which additionally may be supported by user interaction to indicate proper source textures or to supervise machine learning processes.

9.2.1 HALLUCINATION

Wang et al. [335] propose a solution for restoring image details, which are clipped in very dark or bright image regions, based on similar textures that are preserved in well-exposed regions. Such textures can be transferred into the clipped regions using texture synthesis methods. Since the precise reconstruction of clipped textures is usually not possible, the aim here is a plausible approximation of their appearance, and for this reason, the authors call their method "HDR hallucination."

The texture transfer from the source LDR image into the reconstructed HDR image is a more involved problem than in traditional texture synthesis due to highly dynamic changes of lighting between the source texture pixels and their destination clipped pixels. To address this problem, Wang et al. separate lighting and texture information based on a common assumption that illumination changes slowly in the scene while textures are mostly represented by high spatial frequency signals. As discussed in Section 8.5.2, bilateral filtering can be used to perform such an image decomposition into the base (lighting) and texture layers, where edge-stopping properties assure that signals associated with structural scene information such as object contours and other strong gradients are fully contained in the base layer [74].

The processing flow in the algorithm proposed by Wang et al. is as follows. First, inverse-gamma correction is applied to the source LDR image to reconstruct a linear radiance map, in which under- or overexposed pixels are then found by simple thresholding. The radiance map is then processed by bilateral filtering, and problematic pixels are hallucinated independently for the base and texture layers.

To recover lighting information in the base layer, an elliptical Gaussian lobe is placed at the centroid of each saturated region, and its shape and orientation are then fitted through an optimization procedure to well-exposed pixels at the boundary of the saturated region. If needed, such reconstructed lighting can be further adjusted through manual interaction.

The texture component is reconstructed within the texture layer using constrained texture synthesis [72], which is guided by user interaction. The user draws a pair of strokes to indicate the source texture and its destination location, where the stroke shape and size control the texture deformation (warping) and determine its scale and orientation.

An example is the stained-glass image in Figure 9.6, which illustrates the texture transfer and warping from the left to the right window. In the bottom gate scene,

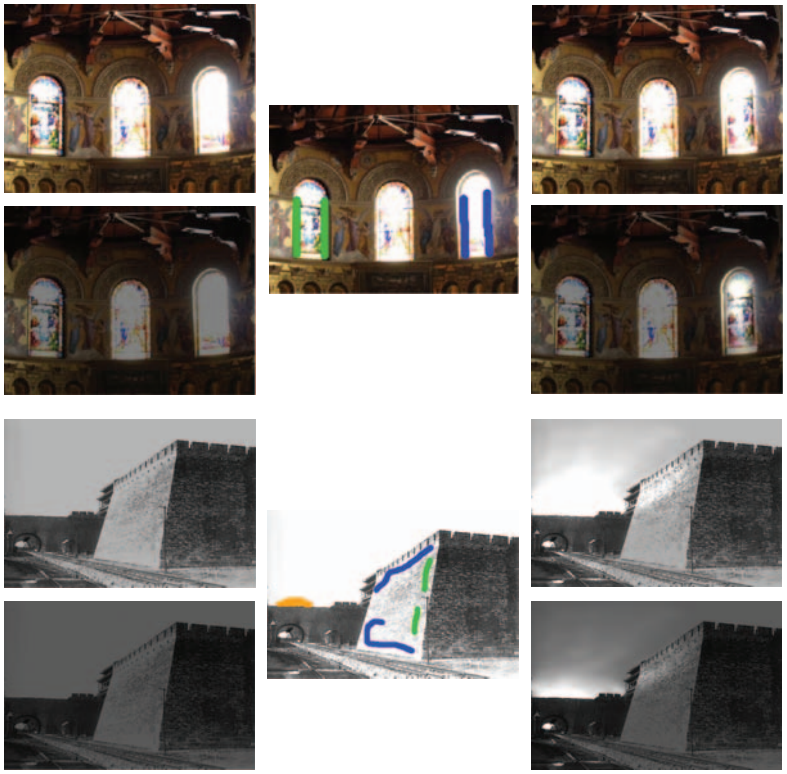


FIGURE 9.6 HDR hallucination results. For each of the two image groups, two different exposures of the original image are shown on the left, and the corresponding HDR hallucination results are shown on the right. The central images illustrate the user interaction, where the green and blue strokes depict the warp tool for the source and destination textures. The orange brush is used to adjust illumination. (Images courtesy of Lvdi Wang, Li-Yi Wei, Kun Zhou, Baining Guo, and Heung-Yeung Shum [335] of Microsoft Research Asia; copyright. The Eurographics Association 2007, used with permission). Pages 329 and 391

apart from the wall texture transfer, interaction with lighting is performed in the region denoted by the orange illumination brush, which affects only overexposed pixels.

At the final stage, the reconstructed lighting and synthesized textures are blended with the original image, and Poisson editing [251] is performed to smooth out possible seams between the hallucinated and the original pixels. While the proposed technique is not fully automatic, in many industrial applications, the possibility of manual control is considered as an advantage. Note that when appropriate source textures cannot be found in the original image, a successful texture transfer can be performed from other images as well.

9.2.2 VIDEO PROCESSING

In off-line video processing, the scenario where user interaction is allowed for selected frames and then such interaction outcome is automatically propagated for the whole video, is perfectly acceptable. Didyk et al. [67] propose a system that is designed along such guidelines. Its main objective is to enhance bright luminous objects in video sequences. First, the system automatically classifies all clipped (saturated) regions into three categories:

Diffuse: Bright (usually white) and partly saturated objects

Reflection: Highlights on shiny strongly reflective surfaces

Lights: Directly visible light sources

The authors have chosen not to enhance diffuse objects, as this is a difficult task and any failures lead to highly visible artifacts. Further, it is likely that the presence of saturation for diffuse surfaces is a conscious decision of the filmmaker, in which case the enhancement would not be appropriate.

Therefore, only reflections and lights are classified based on eight features. The most important of these are the mean luma of a frame, the local luma median, skew, standard deviation in a clipped region, the region's similarity to a disk, and its major axis ratio. These features performed the best for a training set consisting of 2000 manually classified regions.

The resulting classification is propagated to subsequent frames along optical flow trajectories, matching classified regions between frames. For newly appearing objects and at scene cuts, the automated classifier proposes to the user an initial classification that may be modified as needed. Figure 9.7 shows a snapshot of the user interface, which is used for this purpose. The user input is then used to update the knowledge database, affording training for the online classifier.

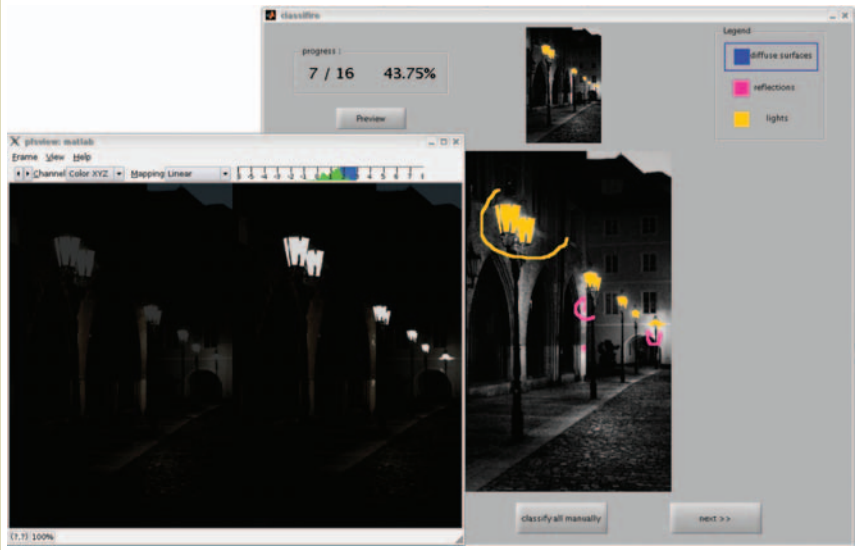


FIGURE 9.7 *Screenshot of the stroke-based user interface for semiautomatic highlight and light source classification as proposed in [67]. With this tool, the user can correct an automatic classification by drawing strokes. Later, changes are propagated to subsequent frames and used to improve the automatic classification. At any time, a preview of the result of enhancement compared with the original frame is available. (Image courtesy of Piotr Didyk)*

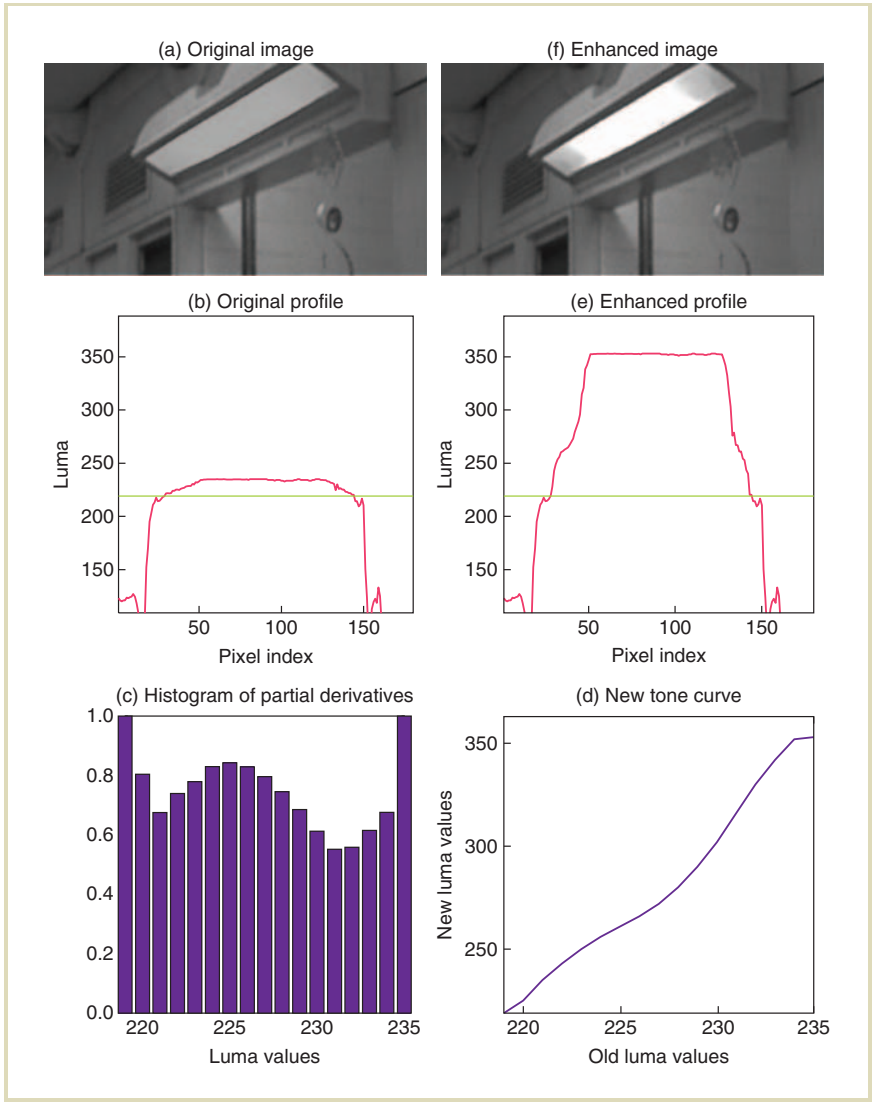
The classified lights are enhanced more strongly than highlights, although the same technique is used for the actual enhancement of pixel intensity. Didyk et al. observe that the fewest artifacts are produced when mostly large gradients are stretched, while small gradients are left intact or are only moderately enhanced. Small gradients are likely to represent noise, and its enhancement is not desirable. On the other hand, large gradients are likely to represent important structural scene information such as objects boundaries, whose enhancement is desirable.

Further, the human visual system is less sensitive to changes in strong contrast due to contrast masking (as discussed in Section 10.7.5), so while brightness enhancement can be achieved, possible artifacts due to contrast stretching should be less objectionable.

The problem then arises of how to build a local tone-mapping curve for each clipped region so that the resulting pixel enhancement follows these objectives. Didyk et al. solve this problem by proposing a histogram-like structure, where they accumulate the information about partial derivatives of neighboring pixel intensities in a clipped region for all frames where this region is visible. For each partial derivative i, which is computed as the forward difference between intensities a_i and b_i ($a_i \neq b_i$), the cost value $|b_i - a_i|^{-1}$ is stored in the histogram for all intensity entries between $a_i + 1$ and b_i. Each entry j in the resulting histogram H is then normalized as $\hat{H}[j] = H[j]/\max\{H[\cdot]\}$ and inverted as $1 - \hat{H}[j]$. In this manner, large values are stored for all histogram entries with contributions from a few large gradients.

Then, by a technique similar to histogram equalization (see Section 8.5.1), the desired tone-mapping curve is built, which is steep for intensities corresponding to large gradients. The process of clipped region enhancement is illustrated in Figure 9.8.

FIGURE 9.8 *Enhancement of a clipped light source region with an adaptive tone curve: (a) initial image, (b) its luma profile, (c) histogram of partial derivatives, (d) tone curve derived from the inverted histogram, (e) its luma profile, and (f) enhanced region. The minimum luma level for clipped regions is indicated in (b) and (e). (Image courtesy of Piotr Didyk [67]; copyright. The Eurographics Association 2008, used with permission). Page 1269*


(a) Original image
(f) Enhanced image
(b) Original profile
(e) Enhanced profile
Luma
Pixel index
(c) Histogram of partial derivatives
Luma values
(d) New tone curve
New luma values
Old luma values

9.3 SUPPRESSING QUANTIZATION AND ENCODING ARTIFACTS

Excessively limited bit depth in LDR images results in loss of low-amplitude details, that is, of errors that are below the quantization threshold. This could potentially be visible on a high-quality display device. Another visual consequence of limited bit depth is *contouring*, which forms false contours (also called "banding artifacts") in smooth gradient regions (see Figure 9.9). Note that such contouring for chromatic channels is often called "posterization" (also shown in Figure 9.9). Dynamic range expansion functions, as discussed in Section 9.1, often lead to such artifacts, in

FIGURE 9.9 *The luminance channel of the image on the top left was reduced in bit depth by 3 bits (from 8 to 5 bits per color channel per pixel), resulting in the top middle. The contouring is especially visible in the sky. Such contouring is also present in 8-bit images but often remains below the visible threshold. However, when the dynamic range of such an image is expanded, contouring artifacts may become suprathreshold and therefore visible. The top-right image was created by reducing the bit depth by 3 bits in each of the three channels, resulting in posterization. The bottom row shows results for a bit-depth reduction by 5 bits. (Photograph courtesy of Tania Pouli)*

particular for image regions where strong contrast compression has been performed due to highly nonlinear tone mapping.

The particular choice of strategy to reduce contouring depends on the availability of the reference high-bit-depth image. When such an image is available, it is possible to encode a low-bit-depth image from its high-bit-depth counterpart so that it survives the image-quantization step and can be recovered at the display stage.

For example, the so-called bit-depth expansion (BDE) technique belongs to this category, where imperceptible spatiotemporal noise is added to each frame before the quantization step is performed. This is similar to dithering, which is routinely used to randomize the quantization error in digital audio and video processing. Intensity averaging in the optics of the display and the human eye then leads to the recovery of information below the quantization step. In this way, a 10-bit visual quality has been reported for an 8-bit signal [50].

In another approach called "companding" [186], low amplitudes and high frequencies in an HDR image are amplified at the compression stage so that they survive the quantization step to the 8-bit LDR image. Since the BDE is a fully symmetric inverted process, the amplified signals are converted back to their initial level in the companded high-bit-depth image. The drawback of this technique is that amplified image details may affect the appearance of the LDR image. It appears that the compander framework has a stronger potential as a tone-mapping tool, and in this context, it is discussed in Section 8.3.5.

In a typical scenario involving legacy content, higher-bit-depth information is not available. Therefore, the BDE and compander techniques are not directly applicable. In this context, several techniques focus on the removal of existing contouring artifacts (decontouring). These techniques include *adaptive filtering*, *coring*, and *predictive cancellation*. Such techniques are often incorporated in modern LCD television sets to enhance DVD quality.

In *adaptive filtering*, contouring artifacts are smoothed out and special care is taken to avoid excessive blurring of useful image information. For example, edge-stopping filters such as bilateral filtering can be tuned to selectively suppress high-frequency and low-amplitude signals. Based on expected contouring contrast, the intensity domain parameters can be precisely selected, while the spatial support can be limited to a few neighboring pixels.

Coring techniques are based on a similar principle but offer more control over high-frequency detail filtering through a multiband image representation [34].

Filtering is applied only to a selected number of high-frequency bands, with its strength gradually decreasing for lower-frequency bands. Note that both adaptive filtering and coring may lead to excessive loss of useful signal that cannot be distinguished from contouring. For example, removing low-amplitude and high-frequency signal from an image representing human skin leads to an unpleasant plastic-like appearance, which is not acceptable.

In *predictive cancellation* [51], the quantization error is estimated based on the low-bit-depth input image, and compensation for this error is then performed prior to image display. First, the low-bit-depth image P undergoes low-pass filtering, which results in a low-spatial-frequency image L with pixels that have higher precision than in P due to averaging (see Figure 9.10). Obviously, this precision gain in L is obtained only for slowly changing signals.

Then, when the quantization operator Q with the same bit-depth accuracy as in P is applied to L, the difference $E = Q(L) - L$ approximates the quantization error inherent for low spatial frequencies in P. Then, by subtracting the error E from P, the most objectionable contouring at slowly changing image gradients is removed. It is assumed that higher spatial frequencies in the contouring signal are less objectionable due to lower sensitivity of the human visual system (see the contrast sensitivity function in Section 10.7.3). This means that higher frequencies can remain in the image. Visual masking can further reduce the visibility of contouring, in particular for high-contrast textures. Section 10.7.5 discusses the concept of visual masking.

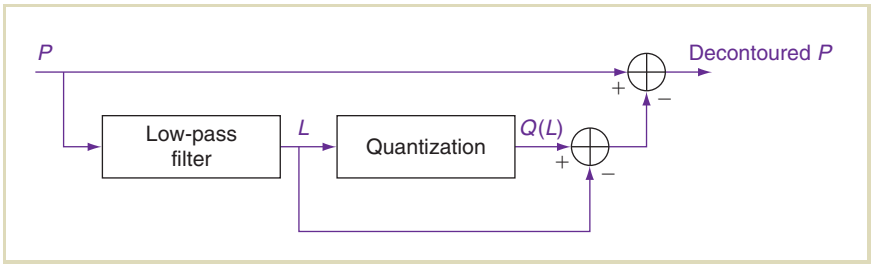


FIGURE 9.10 *Predictive cancellation. The decontoured image P is submitted to a display device.*

9.4 PREFERENCE STUDIES

Most dynamic range expansion works on the premise that some values in the LDR input should be expanded more than other values. For instance, highlights can be expanded more than darker values. However, we can ask ourselves whether such approaches are in any way favored by the human visual system. To date, two such studies have appeared, the first working on the premise that the material to be expanded is well-exposed and the second expanding upon this notion by incorporating images that are not well-exposed.

9.4.1 WELL-EXPOSED IMAGES

Akyüz et al. devised a psychophysical experiment based on user preferences [6]. To evaluate the effects of dynamic range and mean luminance, a set of 10 HDR images were constructed, shown in Figure 9.11. These images were captured such that they did not contain under- or overexposed regions, and they were calibrated to contain absolute luminances. These HDR images were assembled from multiple exposures and could be displayed directly on an HDR display.

For each scene, the subjective best exposure was also selected as a stimulus. The subjective best exposure was selected by means of a pilot study in which participants were asked to select the exposure they preferred. This LDR image was subjected to an inverse tone-reproduction operator to mimic its counterpart HDR image (which served as ground truth). This exposure was subsequently matched to the HDR to have the same mean luminance. This was achieved by matching gray card values. A one third stimulus was created using the subjective best exposure with a higher mean luminance.

Three further stimuli were created, based on the subjective best exposure. In each case, the dynamic range was expanded to yield a higher mean luminance. The expansion was achieved by means of a gamma-like exponent:

$$L' = k\left(\frac{L - L_{\min}}{L_{\max} - L_{\min}}\right)^{\gamma} \tag{9.3}$$

FIGURE 9.11 The test scenes used in the assessment of various dynamic range expansion approaches. These images were tone mapped using the photographic tone-reproduction operator discussed in Section 8.1.2. (Photographs courtesy of Ahmet Oğuz Akyüz)

The three stimuli were created by choosing different values for γ:

$\gamma = 1$. This is effectively linear scaling and constitutes the simplest way to create an HDR image from an LDR image.

$\gamma = 0.45$. Here, a nonlinear curve is created, which amplifies light pixels with respect to dark pixels.

$\gamma = 2.2$. Conversely, in this case, dark pixels are made relatively lighter than the lightest pixels.

Figure 9.12 shows the mappings of these three curves. The reason these curves were chosen is that no assumptions were made as to the correct way to achieve dynamic range expansion: the two nonlinear curves lie on either side of the linear mapping.

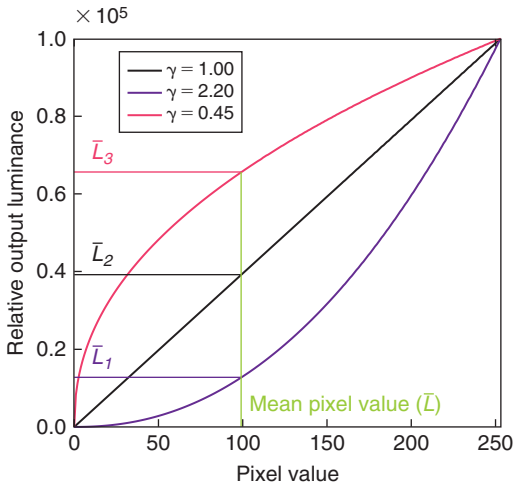


FIGURE 9.12 *The dynamic range expansion curves used in Akyüz's experiment.*

Thus, there were six different stimuli per scene, with 10 scenes total. For each scene, the six stimuli were ranked according to five different criteria, which were general preference, naturalness, visual appeal, spaciousness, and visibility. These criteria allowed more detailed conclusions to be drawn as to why some stimuli may be preferred over others. Ranking was performed by 16 participants, with an even gender distribution. The mean ranking for each criterion and each stimulus is shown in Figure 9.13. Shorter bars mean a higher ranking.

The main conclusion that can be drawn from this figure is that linear scaling of the subjective best exposure results in the highest ranking for all criteria, except naturalness. Direct display of the HDR image was ranked equally with the linearly scaled image in terms of naturalness.

The first three bars of each graph in Figure 9.13 can be compared to assess the effect of dynamic range and luminance on visual preference. Generally, the lighter subjective best exposure is preferred to the direct display of the HDR image on an HDR display. At the same time, the direct display of the HDR image is preferred over the subjective best exposure with the same mean luminance. This result suggests that most participants favor brightness. When the mean luminance is the same, then participants favor images with a higher dynamic range.

The rightmost three bars in each plot compare the different types of inverse tone mapping explored in this study. Here, it can be seen that linear scaling is preferred over either of the two nonlinear expansion algorithms. This is an interesting result, given that conventional wisdom has it that some form of nonlinear scaling is in some way better than linear scaling. In particular, it is generally assumed that light pixels should be expanded more than darker pixels. This study shows that human preferences do not necessarily correlate with this general approach.

This study, however, does not compare visual preference over prolonged exposure times. It is possible that long exposures to very bright displays lead to a preference for images with a much lower mean luminance. In those cases, it may well turn out that a nonlinear expansion function is favored that expands the bright parts of the scene more than the dark parts.

9.4.2 UNDER- AND OVEREXPOSED IMAGES

Whether user preferences remain the same if under- and overexposed images are expanded for display on an HDR display device is the subject of a second user

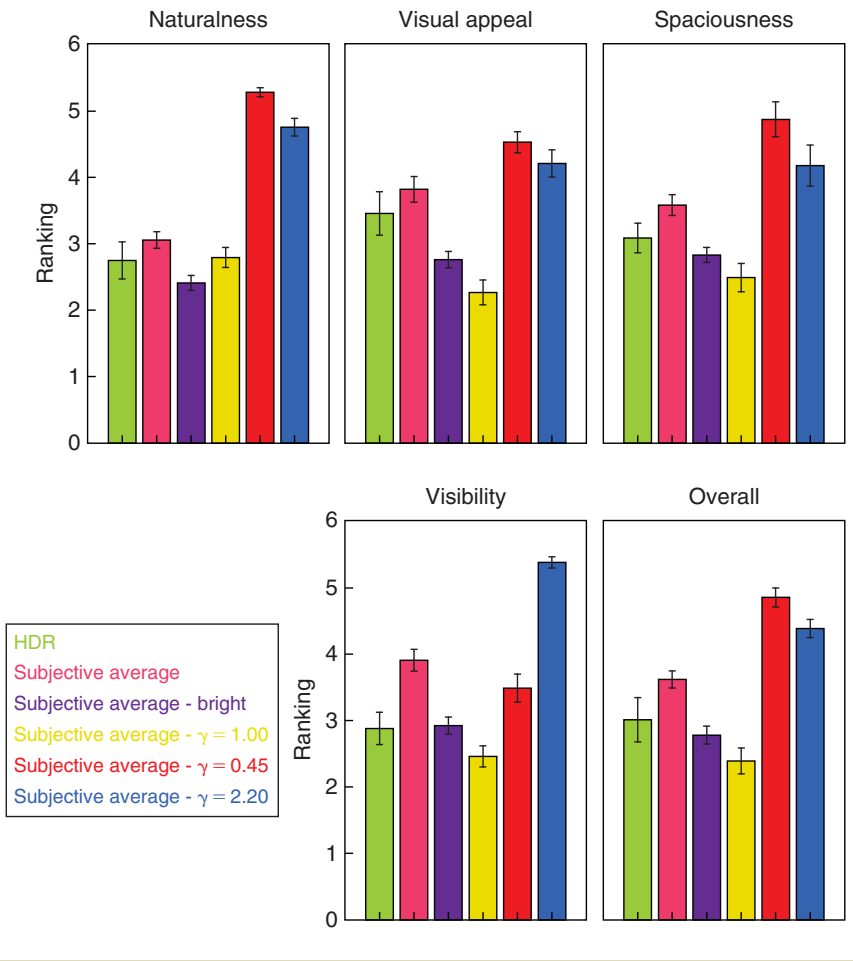


FIGURE 9.13 *Mean rankings for each of the five visual attributes. The error bars indicate the standard deviation of the mean. Lower values indicate a better ranking.*

study [213]. To this end, images were captured in a sequence with nine different exposures. These exposures were then expanded to HDR by one of four algorithms, namely Banterle's approach (Section 9.1), the LDR2HDR algorithm (Section 9.1.4), linear upscaling (Section 9.4.1), and an image matched to the range of a typical desktop thin film transistor monitor. The latter stimulus was added to assess whether HDR imagery is preferred over LDR imagery.

As these four stimuli are derived for each of the nine exposures of the exposures sequence (and for a range of different images), each algorithm can be assessed as to how well it handled under- and overexposure. In a quality-rating procedure, participants viewed the four renditions and selected the visually best image. The exposures were subdivided into an overall lighter and an overall darker group, each group having four exposures. The mean ratings are shown in Figures 9.14 and 9.15.

The results indicate that underexposed images are more amenable to dynamic range expansion than overexposed images. Comparing the different inverse tone-reproduction operators, it was found that on average the LDR2HDR and linear scaling approaches were preferred over Banterle's algorithm. Statistically, the first two algorithms could not be distinguished. In essence, the results shown here are in concordance with those found by Akyüz et al. (described in the preceding section).

Finally, Masia et al. [213] use their results to develop a new algorithm, based on gamma expansion. The key k of the image is estimated based on the image's minimum, maximum, and average luminance [8]:

$$k = \frac{\log L_{\text{avg}} - L_{\text{min}}}{L_{\text{max}} - L_{\text{min}}} \tag{9.4}$$

after excluding 1% of the darkest and lightest pixels. The γ value for the expansion is assumed to linearly depend on the key of the scene, i.e.:

$$k = a\gamma + b \tag{9.5}$$

In a pilot study, the constants a and b were determined to be optimally chosen as $a = 10.44$ and $b = -6.282$. The inverse tone-reproduction operator is thus obtained with

$$L' = L^{\gamma} \tag{9.6}$$

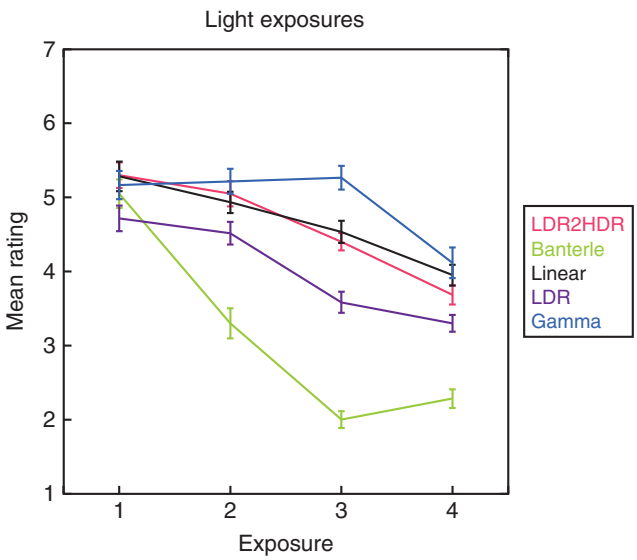


FIGURE 9.14 *Four of the light exposures, inverse tone mapped with one of four algorithms, and rated by the participants. Shown here are the mean ratings.*

This specific approach was also subjected to psychophysical experimentation, the results of which are shown in Figure 9.14. The ratings for this technique for over-exposed material are higher than for the other techniques, suggesting that for those images, gamma expansion may be a useful approach.

Finally, Masia et al. show that the gamma value depends on the number of over-exposed pixels. When the images approach the case where they are well-exposed, the gamma value approaches 1, yielding a result that can be seen as consistent with, and an extension of, Akyüz's proposed linear scaling.

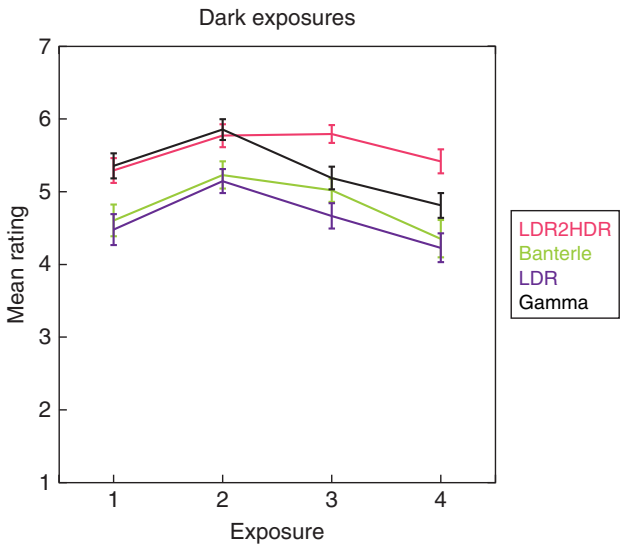


FIGURE 9.15 *Four of the dark exposures, inverse tone mapped with one of four algorithms, and rated by the participants. Shown here are the mean ratings.*

9.5 SUGGESTED APPLICATIONS

Recently, some applications have started to appear that rely on HDR images as input. The first and foremost example is image-based lighting, which is used in the special-effects industry and is discussed in Chapter 11. A second example is an image-based material editing of Khan et al. [155]. Here, an object depicted in an image is first manually selected (i.e., an alpha matte is created for the object), after which the object's material is replaced with a new object. An example is shown in Figure 9.16.

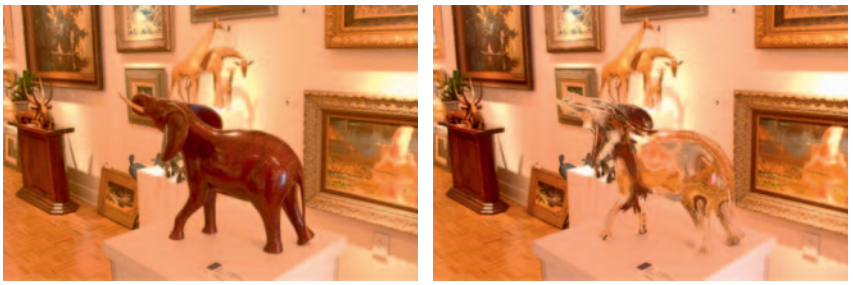


FIGURE 9.16 *An example of image-based material editing. Here, the left image is manipulated to construct the impression of transparency on the right. These results rely on the input image being captured in HDR.*

The reason that this application requires HDR input is twofold. First, the object's geometry is estimated from its shading. Shape from shading algorithms that aim to achieve such tasks tend to rely on the estimation of surface normals, which by themselves are computed from image gradients. In turn, image gradients computed from 8-bit quantized data tend to be imprecise due to quantization. Even if the quantization is below visible threshold, the resulting image gradients may end up being to coarsely quantized to be effective in image-processing tasks.

Second, the background pixels are eventually used to relight the object after insertion of a new material. Such relighting can only be done effectively with HDRI techniques, as shown in Chapter 11. If the input is an LDR image, the creation of an approximate environment map will require the estimation of HDR data from an LDR input image.

This particular problem also appears in webcam clip art, a collection of techniques to use webcams to collect imagery that is then used to transfer the appearance and illumination from one scene to another [172]. Here, environment maps have to be estimated for the same reason as in the work of Khan.

For conventional LDR images to be useful for this type of application, their dynamic range should be expanded. In particular, the effects of quantization should be removed. Whether the techniques described in this chapter can be used to prepare LDR images for applications in image processing is currently an interesting open question.

9.6 SUMMARY

With the advent of HDR display devices, it becomes important to prepare LDR images for such displays. Robust on-the-fly solutions as presented in Section 9.1.4 are of particular importance, since they can be embedded in new generations of displays and tuned to obtain the best performance for a given display type. Such solutions enable the enjoyment of HDR content without waiting for painful format standardization and broadcasting HDR-enabled video signals. However, such dynamic range enhancement is an ill-posed problem in which precise reconstruction of the original HDR content is difficult and often not possible. Fortunately, in the vast majority of applications, a visually plausible appearance of reconstructed HDR material is sufficient. An exception is perhaps image processing, where under some circumstances the correct reconstruction of unquantized data may be required.

Finally, we assumed in this chapter that dynamic range expansion is along the dimension of luminance, that is, that the gamut of the display is extended in only one dimension. However, display manufacturers are equally experimenting with using more saturated primaries, which lead to extended color spaces in all three dimensions. It may, therefore, become desirable in the future to enhance the color gamut of conventional images to match the wider gamut of forthcoming displays.

Visible Difference Predictors

10

Image-quality evaluation is important in many applications, from image acquisition, synthesis, and compression to restoration, enhancement, and reproduction. Quality metrics are commonly used in algorithm validation and comparison tasks. For example, to speed up rendering computation, simplifying assumptions are often used in designing novel algorithms, and quality metrics can be used to determine whether this leads to visible image distortions or artifacts. A typical application of quality metrics is the evaluation of various algorithms' efficiency in terms of texture, image, and video compression. Another group of quality metric tasks is the optimization of algorithms in terms of visual quality. Here, quality metrics are often embedded into control loops, and the goal is to minimize perceivable error. In this case, quality metrics can be used to determine optimal values of certain parameters or to decide upon stopping condition when further computation does not lead to any visible changes in the resulting images.

The problem of quality evaluation is well covered for low dynamic range (LDR) images and video in a number of relatively recent textbooks [367,339,370]. A natural question that arises is whether we need similar metrics for high dynamic range (HDR) images. One may argue that a vast majority of image reproduction devices and media are still of LDR; thus, the human observer deals at the end mostly with LDR images, even if imaging pipelines are of full-fledged HDR. The first observation here is that modern liquid crystal displays (LCDs) and plasma displays are

© 2010, Elsevier Inc. All rights reserved.
DOI: 10.1016/B978-0-12-374914-7.00010-5

much brighter than the cathode ray tube (CRT) displays for which many of the existing quality metrics have been designed. Clearly, higher sensitivity to contrast for brighter displays should be accounted for. Also, some applications may require a visibility measure for real-world conditions in order to understand what can really be seen by a human observer in terms of contrast for a given adaptation luminance level. The detail visibility in such conditions could be compared to those reproduced on the display of a drive or flight simulator. This application would not only require HDR capabilities of a quality metric but *dynamic range independence* that would be in place as well to compare the real-world HDR against tone-mapped LDR images. Many imaging and rendering systems use physically accurate luminance information in the form of HDR images, textures, environment maps, and light fields to capture accurate scene appearance. For such systems, it may be important to compare directly the quality of HDR images at intermediate processing stages. From this short discussion, with the development and proliferation of HDRI technology, adequate quality metrics must catch up to ensure the image quality at all processing stages from acquisition to display.

In this chapter, we focus on quality issues specific for HDR images and the problem of handling images of drastically different dynamic ranges. First, we briefly overview relevant LDR techniques, and then we show how they can be extended to handle HDR images as well.

10.1 SUBJECTIVE VERSUS OBJECTIVE QUALITY METRICS

Human observers are very good in immediate judgment of image quality, even without any reference image. Therefore, unsurprisingly, subjective methods of image-quality evaluation with human subjects remain the most reliable. To reduce the influence of individual preferences, expectations, and past experiences, the quality rating for a larger number of subjects (over 15 is usually recommended) is averaged in the form of mean opinion score. The subjective factors can be reduced by careful design of experimental procedure, precise instructions, and pilot sessions offered to the subjects. For small distortions whose magnitude is close to the detection thresholds, well-established methodology developed in psychophysics

with strong backing in statistics and signal detection theory can be used [80]. In many practical applications, such as low-bandwidth image and video transmission, the distortions are clearly visible (suprathreshold), so the goal is to estimate their magnitude and impact on perceived quality degradation. To obtain more predictable distortion measures, the International Telecommunication Union (ITU) issued recommendations that not only standardize test procedures and obtain data evaluation methods but also specify subject selection criteria, stimuli characteristics, and viewing conditions [46]. For example, the recommendations ITU-R Rec. BT.500-11 (issued in 2002) and ITU-T Rec. P.910 (1999) have been proposed for video quality evaluation in television and multimedia applications, but some of the presented procedures are suitable for static images as well.

In general, the subjective methods are costly, time-consuming, and demanding in terms of the experimenter knowledge and effort required to produce meaningful quality estimates. For these reasons, many practical applications refer to automatic algorithmic methods, which are objective and thus are not affected by mood, expectation, past experience, or other factors inherent to human subjects. Unfortunately, knowledge of the human visual system (HVS) is still relatively limited, and existing computational models predict well only isolated perceptual characteristics that are responses to rather simple stimuli like sinusoidal gratings. Application of these models in the context of complex images is still problematic, which in turn makes existing objective methods less reliable than subjective alternatives based on such models. However, if carefully selected and calibrated for a given task, objective methods usually lead to acceptable results, which in contrast to subjective methods are always perfectly repeatable for the same stimuli. Often, deep understanding of algorithms behind objective methods is not required, and one can use them in a "black-box" manner. The objective metrics are in fact the only viable option in continuous quality monitoring required, for example, in multimedia streams and television signal broadcasting. Also, quality metrics can be directly embedded into the control loops that optimize two-dimensional image processing and three-dimensional rendering techniques with respect to image content, display device capabilities, ambient light characteristics, and other factors [24,231,264,202,17]. Finally, objective methods are the obvious choice in applications dealing with an overwhelming number of images and video sequences, which are intractable for subjective methods.

10.2 CLASSIFICATION OF OBJECTIVE QUALITY METRICS

Objective image-quality metrics are often classified with respect to the amount of information on the reference image that is required for the evaluation of the distorted image quality as [339,368]

- **Full-reference (FR)**, where the full-reference image is available. This case is typical in image compression, restoration, enhancement, and reproduction applications.
- **Reduced-reference**, where a certain number of feature characteristics for the image are extracted and made available as reference through an additional transmission channel (called also the back-channel) with reduced distortion. This is a typical scenario for transmission and broadcasting applications where the availability of FR on the client side is out of the question. To avoid the back-channel transmission, such feature characteristics and low-magnitude signals, such as water-marking, are directly encoded into the image. The distortion of these signals is then measured after the image transmission on the client side.
- **No-reference (NR)**, where it focuses mostly on detecting distortions that are application-specific and predefined in advance, such as blockiness (typical for discrete cosine transform [DCT] encoding in JPEG and MPEG), ringing, and blurring (typical for wavelet encoding in JPEG2000). Recently, some attempts of NR image contrast, sharpness, color saturation, and presence of noise evaluation have been reported; however, combining all these factors into a meaningful quality prediction is a difficult problem [35].

From this point on, we focus on the FR image-quality metrics, as they are better understood and their predictions are usually more reliable. Also, the availability of the reference HDR image is usually not a problem in the applications considered in this chapter.

10.3 FR QUALITY METRICS

The FR quality metrics can be split into three different categories with respect to how the distortion is actually measured and to what extent the HVS modeling is involved to evaluate the distortion perceivability:

- **Pixel-based metrics**, with the mean square error (MSE) and the peak signal-to-noise ratio (PSNR) difference metrics as the prominent examples (Section 10.4). In such a simple framework, the HVS considerations are usually limited to the choice of a perceptually uniform color space, such as CIELAB and CIELUV, which is used to represent the reference and distorted image pixels.
- **Structure-based metrics**, with the *structural similarity (SSIM) index* [338] as one of the most popular and influential quality metrics in recent years (Section 10.5). Since the HVS is strongly specialized in learning about scenes through extracting structural information, it can be expected that the perceived image quality can be well approximated by measuring SSIM between images.
- **Perception-based fidelity metrics**, which are the central part of this chapter (refer to Section 10.6 for an overview) with the *visible difference predictor* (VDP) [49] and the *Sarnoff visual discrimination model* (VDM) [194] as the prominent examples. These contrast-based metrics are based on advanced models of early vision in the HVS and are capable of capturing just visible (near threshold) differences (refer to Sections 10.7 and 10.8) or even measuring the magnitude of such (suprathreshold) differences and scale them in just-noticeable difference (JND) units (refer to Section 10.9).

The selection of discussed metrics has been guided by their suitability to handle HDR image pairs. Some of these metrics, such as MSE, PSNR, and SSIM, can be directly used for this purpose, but more meaningful results can be obtained after their slight modifications. The perception-based metrics such as the VDP and VDM require more profound extensions of visual models to accommodate the full luminance and contrast ranges that can be seen by the human eye. In the following sections, we discuss briefly all these metrics along with their HDR extensions.

In our discussion, we assume that the quality of scene-referred images is directly evaluated. However, the metrics presented here usually require scaling all pixels into photometric luminance values (refer to Section 2.2). Effectively, this means that the output-referred quality of images displayed on arbitrary output device can be easily computed if reproduced pixel luminance values are known. For this purpose, the display transfer function (essentially gamma curve) and its minimum and maximum luminance values must be known. We also assume that if HDR images

have been originally represented in spectral radiance units, they must be converted into luminance to account for the eye spectral sensitivity (refer to Section 2.2).

In many applications, it is considered that the quality evaluation for the achromatic channel is sufficient to capture a vast majority of distortions. Moreover, since the HVS is significantly more sensitive to achromatic contrast (at least five times in terms of detection thresholds), it is often assumed that even if distortions are introduced to both achromatic and chromatic channels, they will be most objectionable in the former case anyway. For this reason, in our discussion, we focus mostly on errors measured for the achromatic channel (luminance), in particular for perception-based metrics such as VDP and VDM. Including color issues into HDR extensions of these metrics is still an open research issue, and even existing LDR solutions [26] are not satisfactory. A notable exception here is iCAM06 [168], which is designed mostly as a color appearance model for complex images, but can be used to measure differences between color HDR images. For simpler MSE, PSNR, and SSIM quality metrics, the achromatic channels can be treated in the same way as the chromatic channel. Of course, the accuracy of obtained quality prediction may suffer as the important color perception effects, such as sensitivity to chromatic contrast [229], chromatic adaptation, and interaction between the achromatic and chromatic channels, such as the Helmholtz–Kohlrausch and Hunt effects, are not modeled [82,168].

10.4 PIXEL-BASED METRICS

The MSE is perhaps the simplest and most commonly used image-quality metric. For a pair of images x and y, MSE can be computed as the mean of squared differences between their pixel values as

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2, \tag{10.1}$$

where N is the number of pixels. The PSNR in decibels is defined as

$$\mathrm{PSNR} = 10\log_{10}\frac{R^2}{\mathrm{MSE}} = 20\log_{10}\frac{R}{\sqrt{\mathrm{MSE}}}[\mathrm{d}B], \tag{10.2}$$

where R is the maximum pixel value, which, for example, takes value 255 for an 8-bit image encoding.

The popularity of these image difference measures is due to their simplicity. Also, the minimization of MSE leads to the familiar least-squares problem solution. In engineering applications, people have built an immediate intuition of what the image-quality reduction by a certain number of decibels (dB) means in terms of expected image appearance (e.g., in the context of image compression). On the other hand, both metrics are based on pixel-by-pixel computation without considering any spatial relation. This means that the original image content has no influence on the error measure (content-agnostic). Thus, adding the same error signal to arbitrary images always leads to the same error prediction, although it is clear that for more textured and high-contrast regions, the distortion is less visible due to signal masking. Similarly, the distortion signal may have different characteristics, for example, high-frequency noise versus localized high-contrast scribbles, which have clearly different visibility but can still lead to similar MSE (distortion-agnostic). Wang and Bovik [339] provide a huge number of spectacular MSE failure examples accompanied with insightful discussion on this topic.

The PSNR metric is well-established in image and video compression applications, and it works particularly well when advanced HVS models are used to control the visibility of compression distortions and their distribution across the image space. The idea behind modern compression algorithms such as JPEG2000 [380] is to hide the distortions imposed by a given bandwidth (the compressed image size restriction) into high spatial frequencies, to which the human eye is less sensitive, and cluttered image regions in which the distortions are more difficult to discriminate due to masking (refer also to Section 10.7.5). In this way, the distortion signal can be perceptually uniformly distributed over the whole image space, and the role of the PSNR metric is limited to detecting its presence and measuring its pixel-by-pixel magnitude.

In the context of HDR images, the question arises that in which units the variables x_i and y_i from Equation 10.1 should be expressed to obtain the most reliable quality predictions in terms of distortion visibility. As this question is not firmly answered in the following section, we discuss a number of possible choices.

10.4.1 PERCEPTUALLY UNIFORM SPACES FOR PIXEL INTENSITY ENCODING

HDR image pixels are often expressed as photometric luminance or its unitless correlates. However, the human eye is not sensitive to absolute luminance values, meaning that by expressing x_i and y_i as luminance, the resulting MSE predictions may correlate poorly with perceivable distortions. One can easily imagine that a luminance difference with respect to a bright background is less visible than the same difference in a dimmer background case (refer also to Section 2.11). As the eye is sensitive to contrast, normalizing the luminance difference by the local background luminance value would immediately result in a better approximation of perceivable differences. This, however, complicates the computational model, as neighboring pixels should be considered to compute the normalization background luminance. In many cases, it is useful to have a perceptually meaningful pixel-by-pixel measure of fidelity. This can be achieved by transforming pixels from the luminance space to a more perceptually uniform space, mimicking the human eye response (perceived brightness) to luminance. In such a space, a unit difference between pixels in both very dark and bright scene regions should be equally perceivable by the HVS. Effectively, this means that such a unit should correspond to a much smaller absolute luminance interval in the dark than in the bright image regions, which in turn requires applying a compressive function to luminance to derive perceptually uniform response. For example, the lightness L^* in the CIELAB and CIELUV color spaces (refer to Section 2.9) fulfill these requirements well, and the respective ΔL^* are commonly used to measure pixel-wise differences between images. For color pixels, CIE ΔE^*_{ab} and ΔE^*_{uv}, or the more recent standard CIEDE2000 [291], can be considered instead of ΔL^*. In fact, the difference measure between gamma-corrected pixel values (refer to Section 2.10), for example, in the sRGB color space, is also a viable solution. (Obviously, the linear RGB space cannot be recommended for the same reason as luminance.)

However, in the context of HDR images, both perceptually uniform CIELAB and CIELUV, as well as sRGB color spaces, require setting the reference white point luminance, which is not trivial for complex images [162]. The choice of brightest pixels usually leads to excessive suppression of details in darker image regions. Better results can be usually obtained by ignoring a certain percentile of the brightest

pixels, but the optimal percentile choice is strongly scene-dependent and difficult to handle in an automatic way. While such a simple approach may work for LDR images, HDR images often present scenes of drastically different luminance ranges, where local adaptation to luminance determines the eye sensitivity (as predicted by the threshold-versus-intensity function discussed in Section 10.7.2). To account for the luminance adaptation, a local reference white as the brightest pixel within a region of radius 50 pixels has been assumed when using the CIE ΔE^*_{94} metric to compare the encoding efficiency for various HDR image formats (Section 3.4). While this procedure leads to meaningful results, the advantage of a pixel-by-pixel quality metric is lost, as spatial processing must be performed within each pixel neighborhood. Also, it is not clear which particular region size should be chosen for local adaptation modeling (the problem of small bright regions) and what the influence of partial adaptation accounting for the reduced eye sensitivity should be [135].

Another problem with the CIELAB, CIELUV, and sRGB color spaces is that they are designed for lower luminance ranges, such as for instance those reproduced by CRT display devices with the maximum luminance often below 100 cd/m^2 in the case of sRGB. For larger luminance values, the power functions with the exponents 1/3 and 1/2.4 used to compress luminance in L^* and RGB channels clearly lead to overestimating the HVS response (refer to Figure 10.1).

A better match to the psychophysical data that model the HVS response to higher luminance levels can be obtained using the logarithm of luminance as predicted by Fechner's law. For comparison purposes, Figure 10.1 illustrates a logarithmic encoding:

$$\text{luma} = 403.8281 \cdot \log_{10}(L) + 1615.4062,$$

which is chosen so that its origin and end point coincide with the JND-scaled plot (refer to Section 10.4.2) for the luminance range considered here, $[10^{-4}, 10^{10}]$ cd/m^2. As can be seen, the logarithmic encoding exaggerates the human eye sensitivity to contrast in dark scene regions, and only for luminance values greater than approximately 100 cd/m^2 Fechner's law holds well. This means that the differences of logarithm-encoded dark pixels lead to a too-conservative image difference prediction with respect to the actually perceived differences (see Figure 10.1).

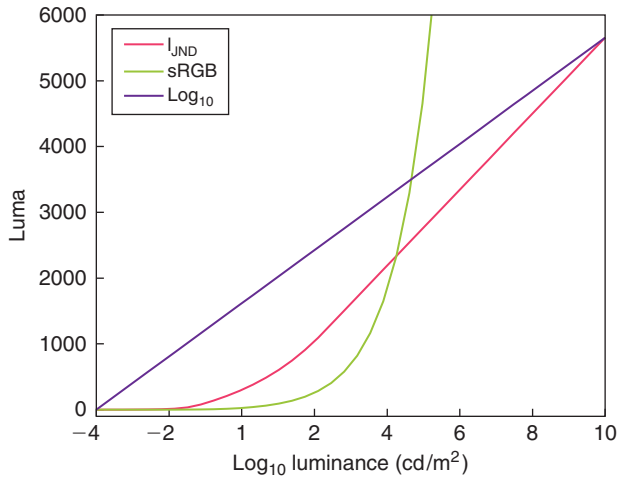


FIGURE 10.1 *Luminance to luma mapping in sRGB, logarithmic, and JND-scaled encodings. (Figure courtesy of Tunç Ozan Aydın)*

10.4.2 JND-SCALED SPACE

In Section 4.1.1, we presented HDR pixel luminance encodings into luma values $l_{\mathrm{HDR}}(L)$ (see Equation 4.4 and Figure 4.2), as well as its inverse function $L(l_{\mathrm{HDR}})$ (Equation 4.5). The main goal of this encoding is to ensure that the difference between the luminance values $L(l_{\mathrm{HDR}}+1)$ and $L(l_{\mathrm{HDR}})$ (i.e., for which the corresponding luma values differ by 1) is a function of visible luminance threshold. Such visible luminance thresholds are predicted by the threshold-versus-intensity (TVI(L)) function and change with luminance adaptation value L (refer to Figure 7.5). In the luma encoding $l_{\mathrm{HDR}}(L)$ designed for video compression, the main goal is to keep the luminance quantization error (due to rounding $\lfloor l_{\mathrm{HDR}}+0.5\rfloor$ to nearest integer) safely below the visibility thresholds. For this purpose, a

multiplier of 1.3811 has been used to rescale the visibility thresholds (TVI(L)) in the $l_{\mathrm{HDR}}(L)$ derivation (refer to Equations 4.2 and 4.3). For the luma encoding $l_{\mathrm{JND}}(L)$, which is intended for the purpose of image difference computation, this multiplier is not needed, which means that

$$L(l_{\mathrm{JND}} + 1) - L(l_{\mathrm{JND}}) = \mathrm{TVI}(L) \tag{10.3}$$

To fulfill this requirement, we need to rescale $l_{\mathrm{HDR}}(L)$ in Equation 4.4 as

$$l_{\mathrm{JND}}(L) = 1.3811 \cdot l_{\mathrm{HDR}}(L) \tag{10.4}$$

This means that the unit difference computed for luma $l_{\mathrm{JND}}(L)$ directly corresponds to 1 JND unit. This way, the obtained MSE differences are directly scaled in the meaningful JND units. The same assumption is taken in and similar luminance encoding has been used for luminance-masking modeling in the perception-based quality metric HDR VDP (refer to Section 10.7.2).

The JND-scaled space has been derived under a very conservative assumption that the human eye is perfectly adapted to each pixel luminance value (this way, any search for the spatial region influencing such an adaptation can be avoided). But this means also that the MSE differences computed for luma l_{JND} might be too conservative, as the human eye at its best only partially adapted to pixel luminances, which effectively means that the HVS sensitivity is substantially lower (refer to Section 10.10).

When using this luma l_{JND} encoding, a precaution should be taken that the JND-scaled space as proposed in Equation 4.4 is derived from the threshold contrast measurement. This is a quite different approach than the direct estimation of the luminance-to-brightness relationship by Stevens and Stevens [305] in their suprathreshold magnitude estimation experiment. The experiment resulted in the so-called Stevens' power law, which has been adopted in CIELAB, CIELUV, and sRGB in their luminance-to-lightness (brightness relative to the reference white) transformations. The consequence of this different background behind the JND-scaled space and the other spaces' derivation is that the resulting MSE measurements in these spaces might be different even for luminance ranges for which they have been originally designed.

Aydın et al. [16] experimented with the JND-scaled space encoding of HDR pixels to compute the MSE differences between HDR image pairs. To achieve backwards-compatibility with the sRGB nonlinearity, they linearly rescaled the JND-scaled space encoding so that the differences between both the encodings are minimized for the luminance range 0.1–80 cd/m^2. This way, the obtained difference values are more meaningful for engineers who used sRGB encoding, at the expense of losing the native for JND-scaled space scaling in the JND units.

From the above discussion, it is clear that MSE (PSNR) can be used for HDR images and has many advantages in terms of computational simplicity, intuitive meaning of measured differences for engineering practitioners, and simple formulation of MSE-driven optimization problems. To get the best possible prediction of differences as perceived by the human observer, HDR image luminance values should be converted to a perceptually uniform space that is the best suited for the actual dynamic range of pixels in the image. The goal of finding such a universal space covering the full range of luminance as seen by the human eye is still an open research question, and the JND-scaled space can be at most treated as the first step in this direction. Obviously, the problem of incorporating chromatic channels into such space must be addressed as well.

Figure 10.2 summarizes the processing flow required to use the MSE metric directly for luminance-scaled HDR pixels. In many practical cases, luminance values may not be precisely known, which is often the case when, for example, multiexposure techniques (refer to Section 5.7) are used to derive HDR pixel values. However, even in such cases, pixel values can just be scaled by a simple linear factor to obtain accurate luminance measures. The factor value can be found once for each camera by performing its calibration when a surface of known luminance is photographed (for more details on photometric camera calibration, refer to Chapter 7.6.1 or [232, Chapter 3.2]). If camera characteristics are unknown, assigning typical values of luminance for given lighting conditions (sunlight, covered sky) usually gives sufficient accuracy in deriving meaningful luminance values.

Meaningful MSE can also be computed for images displayed on arbitrary devices, including HDR displays. As shown in Figure 10.3, at first, pixel values are converted into displayed luminances using the display response function. The next processing step is identical as for HDR pixels directly expressed in luminance (see Figure 10.2).

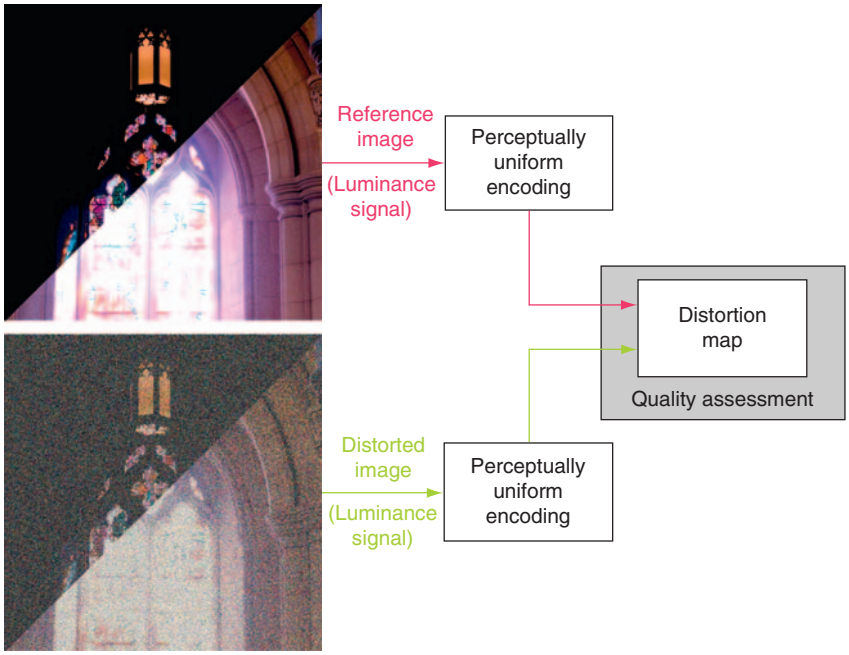


FIGURE 10.2 *Visible difference computation for an HDR image pair with pixel values expressed in luminance units cd/m^2. JND scaling of luminance values is performed before a generic difference metric such as the SSIM index or MSE is used to generate the difference map. (Figure courtesy of Tunç Ozan Aydın)*

It is worth noting that the pixel transformation schemes as shown in Figures 10.2 and 10.3 are generic, and any quality metric can be considered, including the SSIM index, which is discussed in the following section.

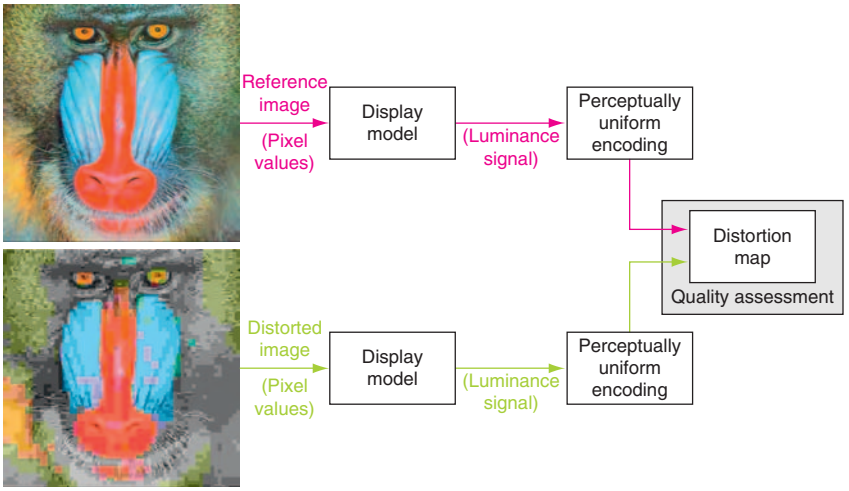


FIGURE 10.3 *Visible difference computation for an LDR (8-bit) image pair. At first, the display model is used to convert pixel values into corresponding luminance values. Then, as in Figure 10.2, JND scaling of luminance values is performed before a generic difference metric is used. (Figure courtesy of Tunç Ozan Aydın)*

10.5 SSIM INDEX

An important trend in quality metrics has been established with the development of the SSIM index by Wang and Bovik [339]. They observe that since the HVS is strongly specialized in learning about scenes by means of extracting structural information, it can be expected that perceived image quality can be well approximated by measuring SSIM between images.

From this standpoint, it is far more important to plausibly reproduce all important image features and preserve the overall image structure than it is to optically match pixels as measured by pixel-based quality metrics. Such features improve the

discrimination and identification of objects depicted in the image, which are important factors in image-quality judgment [140]. The processed image structure can be affected by introducing visible artifacts such as blur, ringing, ghosting, halos, noise, contouring, and blocking [377], which distort the structure of the original image and may degrade the overall impression of image quality.

The SSIM index decomposes similarity estimation into three independent comparison functions: luminance, contrast, and structure.

The luminance comparison function $l(x, y)$ for an image pair x and y is specified as

$$l(x, y) = l(\mu_x, \mu_y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \tag{10.5}$$

where μ_x and μ_y denote the mean intensity:

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{10.6}$$

and N is the number of pixels. The constant C_1 is added in the denominator to avoid the division by zero, and the value $C_1 = (K_1 \cdot D_r)^2$. Here, D_r is the dynamic range of the pixel values and $K_1 \ll 1$. For 8-bit grayscale images, $D_r = 255$ and a good fit to subjective image-quality scores has been obtained for $K_1 = 0.01$ [337]. The $l(\mu_x, \mu_y)$ function is qualitatively consistent with Weber's law by being sensitive to the relative (rather than absolute) luminance change [339].

The contrast comparison function $c(x, y)$ is specified as

$$c(x, y) = c(\sigma_x, \sigma_y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, \tag{10.7}$$

where σ_x and σ_y denote the standard deviation as an estimate of image contrast:

$$\sigma_x = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2} \tag{10.8}$$

The role of constant C_2 is similar to the role of C_1 in Equation 10.5, and a good fit to the just-mentioned perceptual data has been obtained for $C_2 = (0.03 \cdot D_r)^2$ [337].

The function $c(\sigma_x, \sigma_y)$ qualitatively stays in agreement with the contrast-masking property in the HVS (see Section 10.7.5) by being sensitive to relative contrast changes rather than absolute ones [339].

The structure-comparison function $s(x, y)$ is specified as

$$s(x, y) = s\left(\frac{x - \mu_x}{\sigma_x}, \frac{y - \mu_y}{\sigma_y}\right) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3}, \tag{10.9}$$

where σ_{xy} denotes the correlation coefficient between x and y:

$$\sigma_x = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu_x)(y_i - \mu_y) \tag{10.10}$$

Note that the structure comparison is performed on normalized signals $\frac{x_i - \mu_x}{\sigma_x}$ and $\frac{y_i - \mu_y}{\sigma_y}$, which makes it less sensitive for local image luminance and contrast.

The three comparison functions are combined in the SSIM index as follows:

$$\text{SSIM}(x, y) = \left[l(x, y)\right]^{\alpha} \cdot \left[c(x, y)\right]^{\beta} \cdot \left[s(x, y)\right]^{\gamma}, \tag{10.11}$$

where $\alpha > 0$, $\beta > 0$, and $\gamma > 0$ can control the relative importance of each component. In practice, the following form of SSIM index is in common use:

$$\text{SSIM}(x, y) = \frac{(2\mu_x \mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{10.12}$$

where it is assumed that $\alpha = \beta = \gamma = 1$ and $C_3 = C_2/2$. The SSIM index is bounded $\text{SSIM}(x, y) \leq 1$ and $\text{SSIM}(x, y) = 1$ when $x = y$.

To obtain a local measure of structure similarity, the statistics μ_x, σ_x, and σ_{xy} are computed within a local 8×8 window that slides over the whole image (see [339] for details such as a weighting scheme for pixel contributions that are required to avoid blocking artifacts). The local measure can be directly displayed in the form of a quality map. Also, a single numeric value accounting for global error can be computed as an average of each local measure contribution.

The SSIM index gained significant popularity in recent years due to its ability to estimate the variation of perceived structural information between images, combined with an easy-to-understand formulation and low implementation complexity.

In the context of HDR images, it is recommended that pixel luminance is at first transformed into a more perceptually uniform space (see also Section 10.4.2).

Then, over such transformed pixels, the SSIM index can be applied as summarized in Figure 10.2. This may potentially require adapting constants C_1, C_2, and D_r for HDR. For LDR images displayed on a known display, a prior conversion of pixel values to luminance using an appropriate display model is additionally suggested (Figure 10.3). In Section 10.8.2, we present a short comparison of the SSIM index with respect to the other HDR image-quality metrics discussed in this chapter.

10.6 PERCEPTION-BASED FIDELITY METRICS

FR image-quality metrics that incorporate models of human vision are considered to be the most reliable in terms of detecting distortions and in particular for evaluating visibility. These metrics are often called "fidelity metrics," as they just measure perceptible differences between a reference and distorted image pair. Their particular strength stems from the fact that by modeling important characteristics of human vision for image perception, fidelity metrics are relatively precise in establishing whether a given distortion is above the visibility threshold and thus can be perceived by a human observer.

Threshold measurements are well-established in psychophysics [80], and the resulting computational models of human vision are able to explain large volumes of experimental data acquired in many independent experiments. From the image-perception point of view, a problematic issue is that all these models have been built for relatively simple stimuli such as grating patterns. They may, therefore, not always generalize well for complex images. In addition, although most models originally have been developed as stand-alone entities, in fidelity metrics, they are cascaded one after another to mimic processes in the visual pathway from the retina to the visual cortex (Figure 10.5). This may cause unexpected interactions among the cascaded models.

While all these reservations should not be ignored, the HVS-based fidelity metrics are clearly more reliable in the prediction of distortions in visibility than the MSE metric, which completely ignores any aspect of spatial vision. Although spatial pixel processing is performed in the SSIM index, the calibration of parameters controlling

this metric to give precise under- or overthreshold categorization of distortions is difficult, even for multiscale incarnations of the SSIM index [336]. The parameters C_1 and C_2 in Equation 10.12 are of an "abstract" nature and do not translate directly into quantities that can be measured in subjective threshold experiments as required for such calibration.

For HVS-based fidelity metrics, psychophysical experiments can be used to tune their behavior. Subjective threshold measurements with human subjects are used to adjust the metric's predictive power. Through standardization efforts such as ModelFest [355], a large set of image stimuli with various patterns of calibrated threshold intensities has been developed for the purpose of metric calibration; see for instance the Web page of the Video Quality Experts Group (VQEG), http://www.its.bldrdoc.gov/vqeg/index.php, which provides numerous links related to quality metric evaluation and validation activities. For example, according to the outcome of a comprehensive study conducted by VQEG in the years 2000–2004, four of six FR quality metrics developed by institutional proponents such as British Telecom, Yonsei University/South Korea Telecom, and others performed significantly better than PNSR.

Another important role of quality metrics is to measure the magnitude of suprathreshold distortions, for instance in case of low-bandwidth images and video compression. Such distortions require different models of human vision to be plugged into the fidelity metrics. In particular, the metric output should be scaled in JND units to report on distortion magnitude in a perceptually uniform and meaningful way. This is a far more difficult task than detection near threshold distortions.

First, psychophysical data measured in suprathreshold magnitude estimation and scaling experiments are usually noisier. Second, the space of measurements is much wider, in particular for HDR quality metrics where the full visible luminance range and large contrast ranges should be considered. As a result, this parameter space can usually only be sampled relatively sparsely. Deriving robust computational models from such sparse and noisy data, given nonlinearities inherent in the HVS, is a difficult task, hindering the development of suprathreshold HDR fidelity metrics.

In this chapter, we focus mostly on threshold fidelity metrics, which are better researched in the context of HDR images, but we also briefly discuss a suprathreshold case. The vast majority of leading threshold and suprathreshold metrics are based on multichannel image decompositions. In the following section, we first

discuss important characteristics of human vision that are amenable to modeling using multichannel approaches. We then present the processing flow in a generic quality metric based on such models.

10.6.1 MULTICHANNEL MODELS

Although substantial progress in the study of physiology and psychophysics has been made in recent years, the HVS as a whole and the higher-order cognitive mechanisms in particular are not yet fully understood. Only the early stages of the visual pathway beginning with the retina and ending with the visual cortex are considered mostly explored [56]. It is believed that the internal representation of an image by cells in the visual cortex is based on spatial frequency and orientation channels [209,353, 363]. The channel model explains such visual characteristics, which are as follows:

- The overall behavioral contrast sensitivity function (CSF). The visual system's sensitivity is a function of the spatial frequency and orientation content of the stimulus pattern.
- Spatial masking. The detectability of a particular pattern is reduced by the presence of a second pattern of similar frequency content.
- Subthreshold summation. Adding two patterns of subthreshold contrast together can improve detectability within a common channel.
- Contrast adaptation. Sensitivity to selected spatial frequencies is temporarily lost after observing high-contrast patterns of the same frequencies.
- Spatial frequency aftereffects. As a result of the eye's adaptation to a certain grating pattern, other nearby spatial frequencies appear to be shifted.

The latter two characteristics are important mostly for temporal sequences of stimuli. Due to these favorable characteristics, the channel model provides the core of most recent HVS models that attempt to describe spatial vision. For this reason, the channel model is the key component in many state-of-the-art perception-based fidelity metrics.

A generic model of channel-based fidelity metrics is shown in Figure 10.4. The reference and distorted images are first converted into a contrast representation, which is often normalized in terms of visibility (i.e., scaled in JND-like units). Channel decomposition is then performed to separate contrast information in the

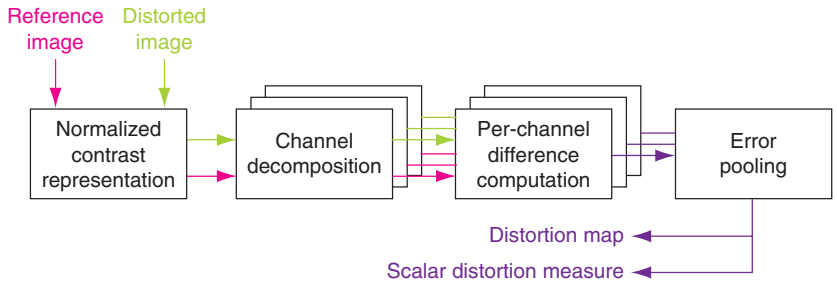


FIGURE 10.4 *Process flow in a generic multichannel perception-based fidelity metric. (Figure courtesy of Tunç Ozan Aydın)*

input images in terms of their spatial location, spatial frequency, and orientation. Figure 10.13 shows an example of such a decomposition. Various filter banks can be used for this purpose, including wavelets [171], steerable pyramid decompositions [320], DCT [354], and the cortex transform [353].

The distortion measure is then computed as the contrast difference between the reference and distorted images for each channel. Often, this measure is additionally normalized to account for spatial masking. Finally, error pooling over all channels is performed to derive the difference map that represents the perceived distortion measure for each pixel. The error in the difference map can further be summed over all pixels to derive a single scalar measure for the pair of images that is being compared.

In the following section, we briefly enumerate the multichannel HDR image-quality metrics that are presented in the remaining part of this chapter.

10.6.2 MULTICHANNEL HDR IMAGE-QUALITY METRICS

The VDP developed by Daly [49] is considered to be one of the leading channel decomposition-based fidelity metrics [183]. It has a process flow similar to the

generic scheme shown in Figure 10.4. In the remainder of this chapter, we present fidelity metrics that are based on the VDP but are also extended specifically for HDR images.

In Section 10.7, we provide an overview of the HDR VDP [203] followed by a more detailed discussion of the choice of computational model of human vision. This choice is guided toward the handling of full visible luminance and contrast ranges typical for HDR images. Such a discussion should not only enable understanding of how the HDR VDP works but also give an insight into the functioning of the HVS itself. These models can be used to build other HDR-enabled visual models, which are suitable to steer rendering and tone-mapping computations [202]. The source code of the HDR VDP, accompanied with detailed documentation, is available under the GPL license and can be accessed on the Internet under the URL http://hdrvdp.sourceforge.net/.

Another quality metric was built based on the same HVS-based models but features quite different characteristics. It is presented in Section 10.8. This *dynamic range independent* metric can directly compare visible structural differences between corresponding HDR and LDR images. Thus, the metric is considered to be a hybrid of SSIM and contrast threshold fidelity metrics. It not only enables the detection of structural changes but also allows the classification of their type and judgment as to whether they can be seen by the human eye or not. Images can be compared using this metric using a free Web service at http://drim.mpi-inf.mpg.de/generator.php.

In Section 10.9, we briefly outline suprathreshold quality metrics that are built in part from HVS models that are also used in the HDR VDP. Finally, in Section 10.10, we discuss the problem of maladaptation and its influence on the prediction of image-quality metrics. We show how the HDR quality metrics discussed in Sections 10.8 and 10.9 can be extended to account for maladaptation.

10.7 THE HDR VISIBLE DIFFERENCES PREDICTOR

The HDR VDP takes as input a pair of spatially registered images, one reference image and one image that has undergone some form of distortion, and produces as output a map of probability values that characterizes whether pixel differences can be perceived. The initial processing is identical for both input images (Figure 10.5).

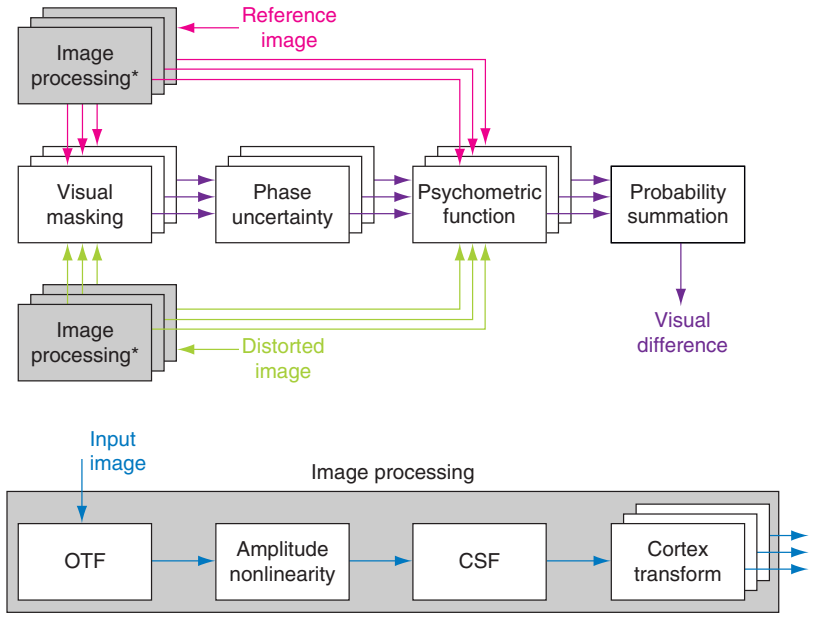


FIGURE 10.5 *Block diagram of the visible difference predictor. Multiple arrows indicate parallel processing of the spatial frequency and orientation channels. (Image courtesy of Rafał Mantiuk)*

It is assumed that all pixel values in the input images represent absolute luminance values.

First, the eye optics is simulated by filtering both images with an optical transfer function (OTF), which essentially models intraocular light scattering responsible for glare effects (Section 10.7.1). Then, the nonlinear response of retinal neurons as a function of luminance adaptation is accounted for. The OTF-filtered pixel values are transformed into the retinal response space using a function derived from

psychophysical measurements of threshold contrast data. The characteristic of the HVS modeled at this stage is often called "luminance masking" (Section 10.7.2).

Then, human visual sensitivity as a function of spatial frequency is simulated using the so-called CSF. It is conservatively assumed that the photoreceptors are perfectly adapted to the local luminance level incident upon them (Section 10.7.3).

The resulting data is decomposed into spatial frequency and orientation channels using the cortex transform, which is a pyramid-style image representation (Section 10.7.4). As a result of the amplitude nonlinearity and CSF computation, the contrast values in all channels are normalized by the corresponding threshold detection values.

However, due to visual masking, those threshold values can be further elevated as a function of contrast in the masking pattern. For every channel and for every pixel, the elevation of the detection threshold is calculated based on the masking pattern's contrast for that channel and that pixel. This means that only intrachannel masking is considered. Conceptually, the distorted image can be seen as the sum of the reference image and a map of distortions. The reference image is therefore usually treated as the pattern that masks the distortion visibility, for example, in cluttered texture regions.

The resulting threshold elevation maps undergo a step modeling phase uncertainty. Essentially, low-pass filtering with a small kernel is performed to account for the relative insensitivity of the HVS to small shifts of the signal phase (Section 10.7.5). The filtered threshold elevation maps are then used to normalize the distortion measure, which is computed as the difference between the reference and distorted images for every pixel and for every channel.

The normalized within-channel distortion measures are then subjected to the psychometric function that estimates the probability of detecting the differences between each channel (Section 10.7.6). These estimated probability values are summed across all channels for every pixel.

Finally, the probability values are used to visualize visible differences between the reference and the distorted images. It is assumed that the difference can be perceived for a given pixel when the probability value is greater than 0.75, which is the standard threshold value for discrimination tasks [363]. When a single numeric value is needed to characterize the differences between images, the percentage of pixels with a probability greater than this threshold value is reported. The probability map

measure is suitable for estimating the differences locally, while the numeric measure provides global information on the perceived differences for the whole image.

It is important to properly interpret these distortion measures, as the HDR VDP is sometimes misused to measure the magnitude of distortions, which is incorrect. The HDR VDP is a threshold fidelity metric whose task is to distinguish between visible (suprathreshold) and invisible (subthreshold) distortions. The metric gives only an indication of the perception of distortions in terms of the probability of detection. When a distortion is strong, the probability of its detection is practically 100% irrespective of its magnitude. For example, the probability of detection does not change significantly between two distortions that are 5 and 20 JNDs. Thus, it is incorrect to infer any strong conclusions regarding the magnitude of distortions on the basis of marginal differences in the probability values when they are close to 100%. If one wants to measure the distortion magnitude, for example, to obtain 5 and 20 JNDs in the given example, a suprathreshold fidelity metric should be used, as discussed in Section 10.9.

In the following sections, we describe the most prominent computational models of human vision that are used in the HDR VDP. These models are important in the perception of HDR images.

10.7.1 EYE OPTICS MODELING

The quality of any image projected on the retina is affected by a number of distortions typical for any real-world optical system, but in particular for the ones built of biological structures. Under normal observation conditions, the distortions lead to a hardly perceivable blur, limiting the occurrence of possible aliasing artifacts that could otherwise arise due to the limited density of retinal photoreceptors. Thus, the ocular media tend to filter frequencies over the Nyquist limit.

Nonetheless, in the presence of directly visible bright light sources, caustics, or highlights, instead of having a crisp projected image on the retina, surrounding regions are affected by scattered light. The following optical elements in the eye contribute strongly to such intraocular scattering (the relative contribution is given in parentheses): the cornea (25–30%), the lens (40%), the vitreous humor (10%), and the retina (20%). The light diffraction on the pupil boundary and eyelashes are additional sources of retinal image-quality deterioration [146]. All these effects

contribute to glare, typical patterns occurring near bright light sources, shown in Figure 8.49. See also Section 8.4.2 for more details on glare rendering using models based on wave-optics principles.

The glare effects, as shown in Section 8.4.2, Figure 8.53, are inherent to the human eye and cannot be captured in an HDR image-acquisition system using standard cameras. Also, due to dynamic range limitations, such effects cannot be invoked in the eye while observing displayed images even on existing HDR displays (see Chapter 6). However, from the standpoint of image quality as seen by a human observer, such effects should be simulated by human visual models in the image fidelity metrics. Obviously, the eye's inability to discern image detail in the proximity of glare sources should be considered.

In practice, only the bloom effect (also known as "veiling glare" and in the visual performance literature as "disability glare") is modeled. This effect is responsible for local contrast reduction and is modeled on the basis of measured point-spread functions for the eye's ocular media [358,303]. Alternatively, it can be represented in frequency space by the OTF [64,210,20]. It is important to perform the bloom computation directly on pixels scaled to absolute luminance values, as intraocular light scattering takes place before retinal adaptation processes are involved. This is particularly important for HDR images for which nonlinearities attributed to retinal processes can significantly distort the outcome of filtering by point spread or OTFs.

In the HDR VDP, the initial luminance map L is filtered using the OTF proposed by Deeley et al. [64]:

$$\mathrm{OTF} = \exp\left[-\left(\frac{\rho}{20.9 - 2.1d}\right)^{1.3-0.07d}\right], \tag{10.13}$$

where d is a pupil diameter in millimeters and ρ is the spatial frequency in cycles per degree (cpd). The pupil diameter is calculated for a global adaptation luminance L_{ga} using the formula due to Moon and Spencer [225]:

$$d = 4.9 - 3 \cdot \tanh\left[0.4 \cdot \log_{10}(L_{\mathrm{ga}} + 1)\right] \tag{10.14}$$

The global adaptation luminance L_{ga} is a geometric mean of the luminance map L (as commonly used in tone mapping — see Section 7.4.1). Since the OTF is defined

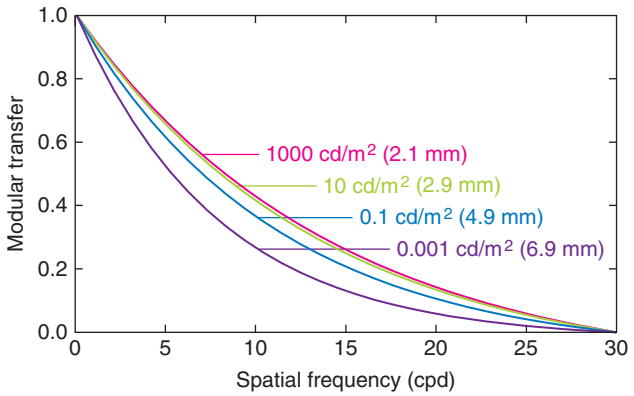


FIGURE 10.6 *The optical transfer function based on Deeley et al.'s model for different adaption luminance levels and pupil diameters (given in parentheses). (Plot courtesy of Rafał Mantiuk)*

in the frequency domain (Equation 10.13 and Figure 10.6), the filtering operation is performed in Fourier space:

$$L_{\mathrm{OTF}} = \mathcal{F}^{-1}\{\mathcal{F}\{L\} \cdot \mathrm{OTF}\}, \tag{10.15}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier transform and its inverse, and L_{OTF} is the resulting optically filtered luminance map, which is used as input to the amplitude nonlinearity stage as described in the next section.

10.7.2 AMPLITUDE NONLINEARITY MODELING

The human eye can operate under a large range of luminance values through adaptation, which is performed at neural and photochemical levels in the retina, as well as through the control of pupil size (for more details on temporal aspects of adaptation, see Section 7.5). The eye is not sensitive to absolute luminance values L but rather responds to relative luminance levels L/L_{a} with respect to a given local adaptation

luminance L_a (this effect is also known as "luminance masking"). However, the eye contrast sensitivity S is significantly lower for the scotopic ranges of L_a, increases for mesopic, and stabilizes at a constant level for photopic conditions. A direct consequence of such variability of the eye contrast sensitivity is an observation that we cannot read a book in dim light, while we can do it easily at daylight levels. In both the cases, the contrast of characters' ink with respect to the background paper is constant irrespective of illumination levels.

Figure 10.7 shows the relation between the threshold contrast $\Delta L/L_a = 1/S$ that can be detected for a given background L_a, which is referred to as the contrast-versus-intensity (CVI) function.[1] Note that $\Delta L/L_a$ constant for $L_a > 100\,\mathrm{cd/m^2}$ as predicted by Weber's law, and the contrast thresholds for lower L_a (in the scotopic and mesopic ranges) are significantly higher. To convert the response R (often called lightness, see Section 2.9) to luminance L, which is compatible with the CVI, a power function for roughly $L_a < 100\,\mathrm{cd/m^2}$ and a logarithmic function for higher L_a should be used [207]. Note that power functions $R = k_1 L^a$ [305] (this relationship is known as Stevens' power law) are commonly used in converting luminance to lightness or to gamma-corrected luma, which are used in perceptually uniform color spaces designed for LDR reproduction devices such as CRT and LCD displays. The logarithmic function is a direct consequence of Fechner's law $R = k_2 \ln L$, which specifies the HVS response to luminance under the assumption that Weber's law $\Delta L/L_a = \mathrm{const}$ holds.

In Section 4.1.1, we discuss a nonlinear response function, originally proposed by Mantiuk et al. [207], which is suitable for the full visible range of luminance. As shown in Figure 4.2, the shape of the response function depends on the CVI function used for its derivation. To make the CVI function (Figure 10.7) compatible with the CSF used in HDR VDP (Section 10.7.3), for each adaptation level L_a, the lowest detection threshold (i.e., the maximum sensitivity over all ρ) is estimated as [203]

$$\mathrm{CVI}(L_a) = \frac{1}{\max_{\forall \rho}[\mathrm{CSF}(\rho, L_a, \ldots)]} \tag{10.16}$$

1 Note that in Figure 7.5, an analogous threshold-versus-intensity (TVI) function is plotted, in which case the ordinal axis represents absolute luminance thresholds rather than contrast thresholds as in Figure 10.7. The TVI plot can be directly converted into the CVI plot through the normalization of absolute luminance thresholds by the corresponding background luminance from the abscissa axis, which is equivalent to deriving contrast thresholds, i.e., $\frac{\mathrm{TVI}(L_a)}{L_a} = \mathrm{CVI}(L_a)$.

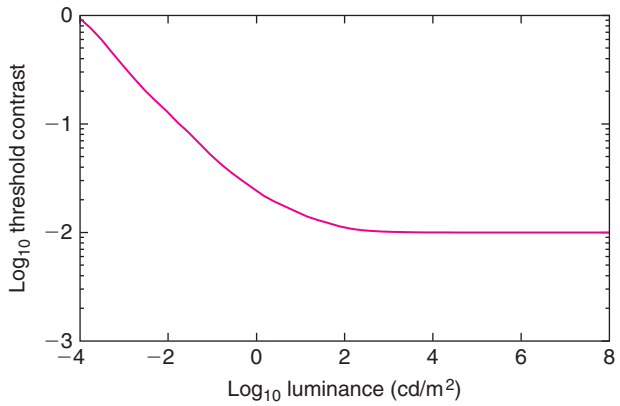


FIGURE 10.7 *The CVI function specifies the threshold contrast value that can be detected for a given adaptation luminance L_a. The CVI function has been derived from the CSF model as presented in Equation 10.17 through searching for a minimum contrast threshold for each adaptation luminance L_a (see Equation 10.16). (Plot courtesy of Rafał Mantiuk)*

Refer to Equation 10.17 for the full list of CSF parameters. The CVI function is then used to derive the luma $l_{HDR}(L)$ mapping (Equation 4.4), which specifies the retinal response $R = l_{HDR}(L)$.[2] As explained in Section 10.4.2, to scale the retinal response precisely in JND units, it is more appropriate to use the JND-scaled space $R = l_{JND}(L)$, which is a rescaled version of $l_{HDR}(L)$ (see Equation 10.4). As can be deduced from Equation 4.4, the JND-scaled space combines properties of a power function for lower luminance values as postulated by Stevens' power law and a logarithmic function for higher luminance values to accommodate Fechner's law.

2 See also Section 10.9.1, which presents a transducer $R = T(c)$ modeling the HVS response to contrast c. In both cases, the HVS sensory responses are modeled, so the retinal response and transducer derivation are based on the same principles (see [362,207,208] for more details), which are applied once to luminance L and second time to contrast c.

While the analytic version of R is convenient to use, its discrete version is actually used in the HDR VDP because of better fitting to the perceptual data underlying the CVI function [203]. In fact, the inverse response function is implemented as a lookup table $R^{-1}[i]$ for discrete values $i = 1, \ldots, N$. The actual photoreceptor response R is found by linear interpolation between a pair of nearest i values corresponding to a particular luminance L_{OTF}.

Figure 10.8 depicts the shape of the response function used in the HDR VDP, which as expected strongly compresses the retinal response for low luminance levels. Thus, the response R to luminance L is guided by the contrast sensitivity for various adaptation luminances L_{a}. It is important to note that in the HDR VDP, perfect local adaptation is assumed, which means that $L_{\mathrm{a}} = L$. In fact, since a light-scattering simulation step (OTF) is preceding the amplitude nonlinearity step in the

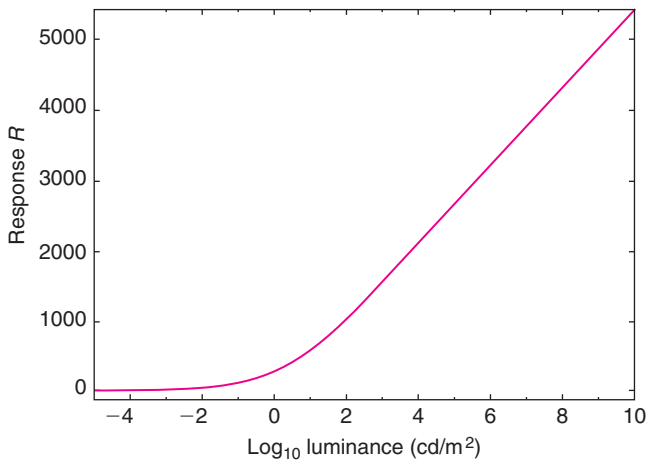


FIGURE 10.8 *The retinal response R, which models the HVS response to luminance represented as the JND-scaled space (see Section 10.4.2). (Plot courtesy of Tunç Ozan Aydın)*

HDR VDP, L_{OTF} derived from the eye optic simulation (Equation 10.15) is the input to the response function, which means that in all derivations $L_{\mathrm{a}} = L = L_{\mathrm{OTF}}$.

10.7.3 CONTRAST SENSITIVITY FUNCTION

The CSF is one of the most important characteristics of the HVS in image perception. As such, it is not surprising that a vast majority of computational visual models include the CSF as their key component.

The CSF has usually been investigated in detection experiments for simple harmonic function gratings, and even with such simple stimuli, many factors can influence the measured sensitivity. Obviously, the gratings' spatial and temporal frequencies, color, and background (adaptation) luminances are the most important factors. However, the CSF also depends on several other factors:

Grating orientation. The HVS has lower sensitivity to oblique patterns.

Spatial extent. Sensitivity goes up with an increasing number of periods and then saturates.

Eccentricity with respect to the fovea. The sensitivity falls off with higher eccentricity.

Observation distance. Lower frequencies become higher with increasing distance. Also, the sharpness of vision reduces with distance.

In the following discussion, we ignore the temporal and colorimetric aspects of the CSF, but an interested reader may refer to [153,152,352,229,154], who treat these problems in their full extent.

In the context of HDR images, it is important that the CSF model is valid for the full range of luminance that can be seen by human observers. For this reason, the CSF model proposed by Daly [49], which is an improved version with respect to Meeteren's CSF [331], can be strongly recommended:

$$\mathrm{CSF}(\rho, \theta, L_{\mathrm{a}}, i^2, d, \chi) = P \cdot \min\left[S_1\left(\frac{\rho}{r_a \cdot r_\chi \cdot r_\theta}\right), S_1(\rho)\right], \qquad (10.17)$$

where

$$
\begin{aligned}
r_{\mathrm{a}} &= 0.856 \cdot d^{0.14} \\
r_{\chi} &= \frac{1}{1 + 0.24\chi} \\
r_{\theta} &= 0.11 \cos(4\theta) + 0.89 \\
S_1(\rho) &= \left[\left(3.23(\rho^2 i^2)^{-0.3}\right) 5 + 1\right]^{-\frac{1}{5}} \cdot A_l \varepsilon \rho \mathrm{e}^{-(B_l \varepsilon \rho)} \sqrt{1 + 0.06 \mathrm{e}^{B_l \varepsilon \rho}} \qquad (10.18) \\
A_l &= 0.801 \left(1 + 0.7\, L_{\mathrm{a}}^{-1}\right)^{-0.2} \\
B_l &= 0.3 \left(1 + 100\, L_{\mathrm{a}}^{-1}\right)^{0.15}
\end{aligned}
$$

The parameters are as follows: ρ, spatial frequency in cycles per visual degree; θ, orientation; L_a, the light adaptation level in cd/m^2; i^2, the stimulus size in deg^2 ($i^2 = 1$); d, distance in meters; χ, eccentricity ($\chi = 0$); ε, constant ($\varepsilon = 0.9$); and P, the absolute peak sensitivity ($P = 250$). Note that the formulas for A_l and B_l have been corrected with respect to the original publication [49]. Figure 10.9 depicts contrast sensitivity for various adaptation luminance values L_{a} as modeled using Equation 10.17.

Before the CSF model specified in Equation 10.17 can be used in the HDR VDP, it should be correlated with the OTF and amplitude nonlinearity steps (Figure 10.5).

While performing contrast-detection experiments, it is difficult to separate the influence of optical scattering and neuronal processing, and for this reason the CSF simultaneously models both of them. The optical component can be easily compensated by dividing the CSF by the OTF. Note that applying the OTF prior to the retinal response computation in the HDR VDP processing, pipeline is required to account for glare effects, which can be very strong for HDR images depicting luminaires, highlights, and other high-contrast features.

The JND-scaled space performs normalization to JND units. As a similar normalization is inherent for the CSF that by definition specifies the contrast threshold equivalent to 1 JND unit, the duplication of such normalization can be avoided by rescaling Equation 10.17 so that it is expressed as a percentage of maximum sensitivity for a given adaptation luminance. This way, the second important feature

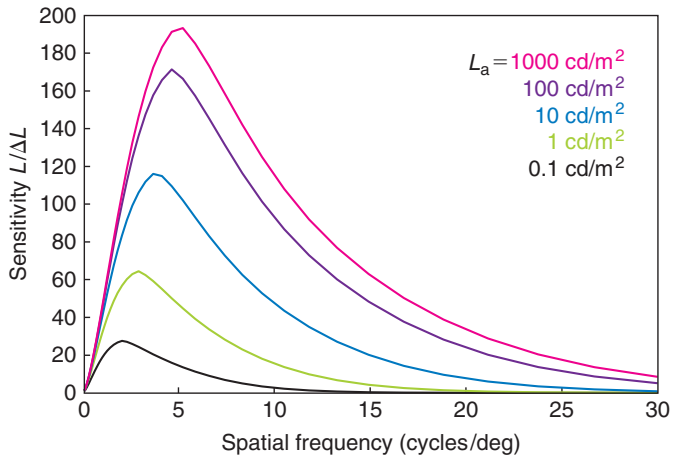


FIGURE 10.9 *The plot of the contrast sensitivity function for various adaptation luminance levels* L_a*, which is derived from Equation 10.17. The sensitivity to low spatial frequencies is increasing with the increase of* L_a*, and then it becomes almost the same for curves* $L_a = 100$ *and* $1000\ \mathrm{cd/m^2}$*. In the region where these curves overlap, the threshold contrast is the same, which means that Weber's law holds. (Plot courtesy of Rafał Mantiuk)*

of the CSF, which is modulating the sensitivity as a function of spatial frequencies, is maintained but scaled in [%] units. In the JND normalization issue, note that the CVI function that has been used in the JND-scaled space derivation is based on the CSF model in Equation 10.17, which can be also interpreted as incorporating the threshold-scaling capabilities of the CSF into the amplitude nonlinearity step.

The effects of optical light scattering and JND scaling can be excluded from the original CSF formulation with

$$\mathrm{nCSF}(\rho, \theta, L_a, i^2, d, \chi) = \frac{\mathrm{CSF}(\rho, \theta, L_a, i^2, d, \chi)}{\mathrm{OTF}(\rho)} \cdot \mathrm{CVI}(L_a) \qquad (10.19)$$

Finally, the normalized contrast sensitivity function nCSF is used to modulate the response values obtained in the amplitude nonlinearity step. Figure 10.10 depicts nCSF for various luminance adaptation levels L_a.

As can be seen in Figure 10.10, for increasing L_a, the peak sensitivity shifts toward higher spatial frequencies, which results in different shapes of nCSF. Since local adaptation L_a is assumed in the HDR VDP, precise nCSF processing could potentially require applying a different filter shape to each pixel. To speed up computations, the response map R is filtered five times assuming $L_a = \{0.0001, 0.01, 1, 100, 1000\}$ cd/m^2, and the final value for each pixel is found by linear interpolation between two prefiltered maps closest to the L_a for a given pixel.

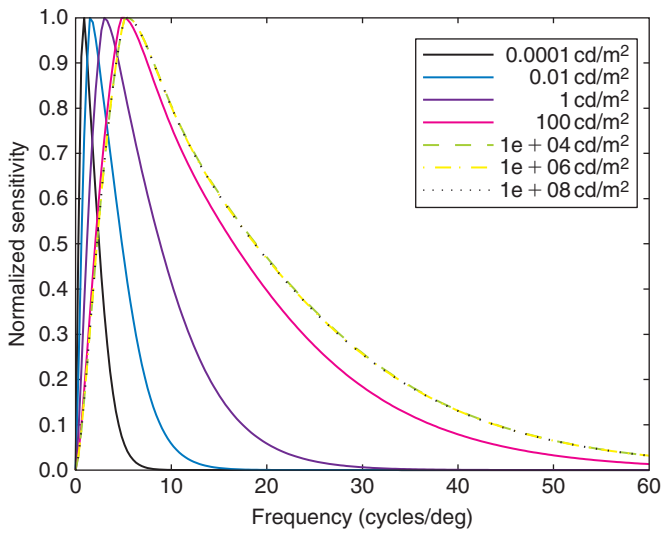


FIGURE 10.10 *The plot of normalized contrast sensitivity function (nCSF) for various adaptation luminance levels L_a. (Plot courtesy of Tunç Ozan Aydın)*

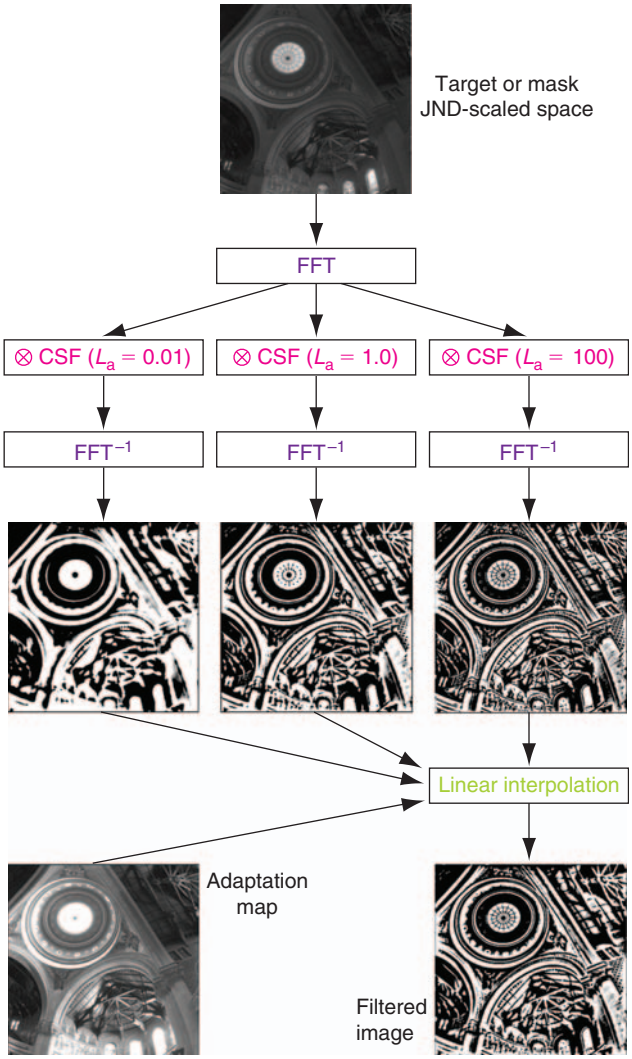
Note that such a limited number of prefiltered maps works much better for nCSF (Equation 10.19) than for CSF (Equation 10.17), in which case the luminance adaptation more strongly affects the resulting filter shapes. Note also that the same filter can be used for $L_a \geq 1000\,\mathrm{cd/m^2}$, as for those values the corresponding nCSF shapes become similar. Figure 10.11 shows the processing flow for nonlinear filtering, which is performed in the Fourier domain. A similar approach is used in Durand and Dorsey's tone-mapping algorithm [74], which also requires nonlinear filtering (Section 8.5.2).

10.7.4 CORTEX TRANSFORM

The HDR VDP follows its LDR predecessor by using a multichannel decomposition technique, modeling the combined radial frequency and orientational selectivity of cortical neurons. The cortex transform is used for this purpose [353,49], which is a pyramid-style, invertible, and computationally efficient image representation. The input image is decomposed into six frequency channels using band-pass filters, and each of the resulting bands is then filtered into six orientation-selective bands (see Figure 10.12). The lowest-frequency baseband is not filtered using the orientation selective filters, so in total 31 bands are obtained. The precise formulation of filter equations used in the HDR VDP can be found in Daly [49] and Aydın et al. [15], while for the original cortex transform formulation as proposed by Watson, the reader is referred to [353].

Figure 10.13 depicts the cortex transform's decomposition of the frequency plane and shows an example of image decomposition for selected bands. Note that the cortex transform filters and the CSF filter (see Figure 10.11 and Equations 10.17 and 10.19) are used in the Fourier domain, while all HDR VDP processing steps

FIGURE 10.11 *Contrast sensitivity processing in the HDR VDP. Since for HDR images local luminance adaptation must be handled, the input image with JND-scaled retinal response is filtered using several shapes of nCSF at selected luminance levels. The final value of a filtered pixel is interpolated based on two precomputed images, which are the nearest to the adaptation luminance for this pixel. (Image courtesy of Rafał Mantiuk)*


Target or mask
JND-scaled space
FFT
⊗ CSF (L_a = 0.01)
⊗ CSF (L_a = 1.0)
⊗ CSF (L_a = 100)
FFT^{-1}
FFT^{-1}
FFT^{-1}
Linear interpolation
Adaptation
map
Filtered
image

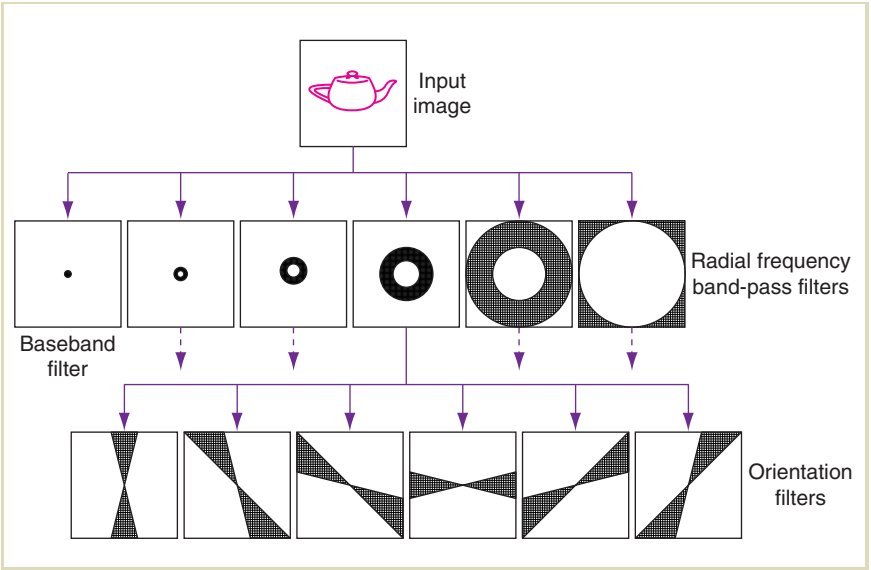


FIGURE 10.12 *The cortex transform: Organization of the filter bank into band-pass and orientation selective filters. The black color denotes spatial and orientation frequencies that can pass through each depicted filter.*

that follow, such as visual masking, phase uncertainty, and psychometric scaling (Figure 10.5), require all channels that are converted back into the spatial domain.

Note that the CSF modeling could in fact be postponed until the cortex transform stage, where multipliers derived from the CSF could be used to scale the contrast signal in each channel [264]. Moreover, the impact of local luminance adaptation L_a on CSF scaling (Figure 10.9) could be easily implemented in the spatial domain. In this case, per-pixel spatially varying multipliers could be derived directly from Equation 10.17 based on the local luminance $L_a = L_{OTF}$ for each given pixel. This could be a way to speed up the HDR VDP computation, but clearly, the overall

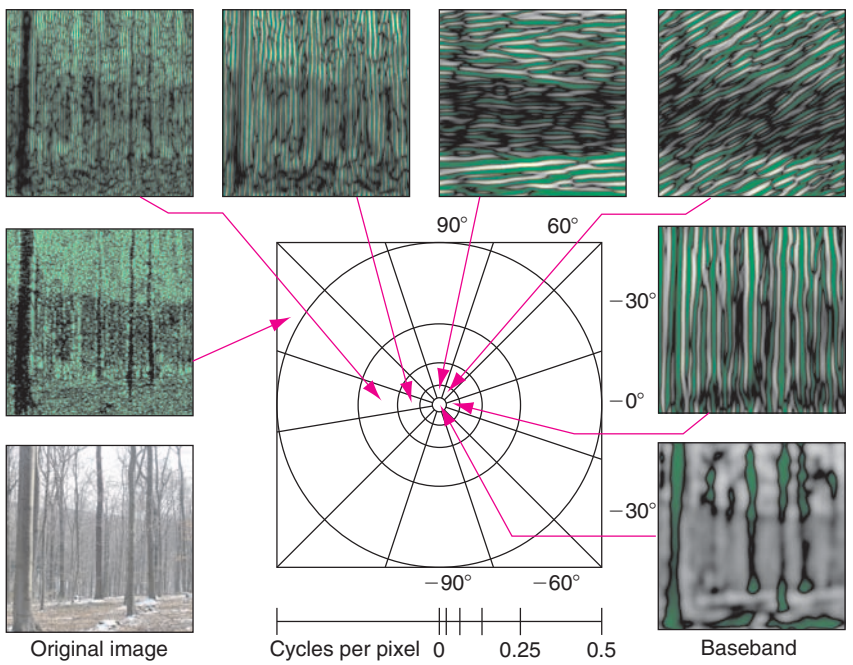


FIGURE 10.13 *Cortex transform: Decomposition of the image frequency plane into radial and orientation selectivity channels. At the bottom, the spatial frequency scale in cycles per pixel is shown, which enables the estimation of the band-pass frequencies of every radial channel. Also, the orientation of the band center in degrees is shown for every orientation channel. The images around show an example of decomposition into the selected band-pass and orientation channels for the original image shown at the bottom-left corner. The shades of gray and green show the positive and negative contrast values, while black denotes zero-crossing or no-contrast regions. (Image courtesy of Rafał Mantiuk)*

accuracy of detection threshold scaling for all spatial frequencies may be too low for any reliable difference prediction.

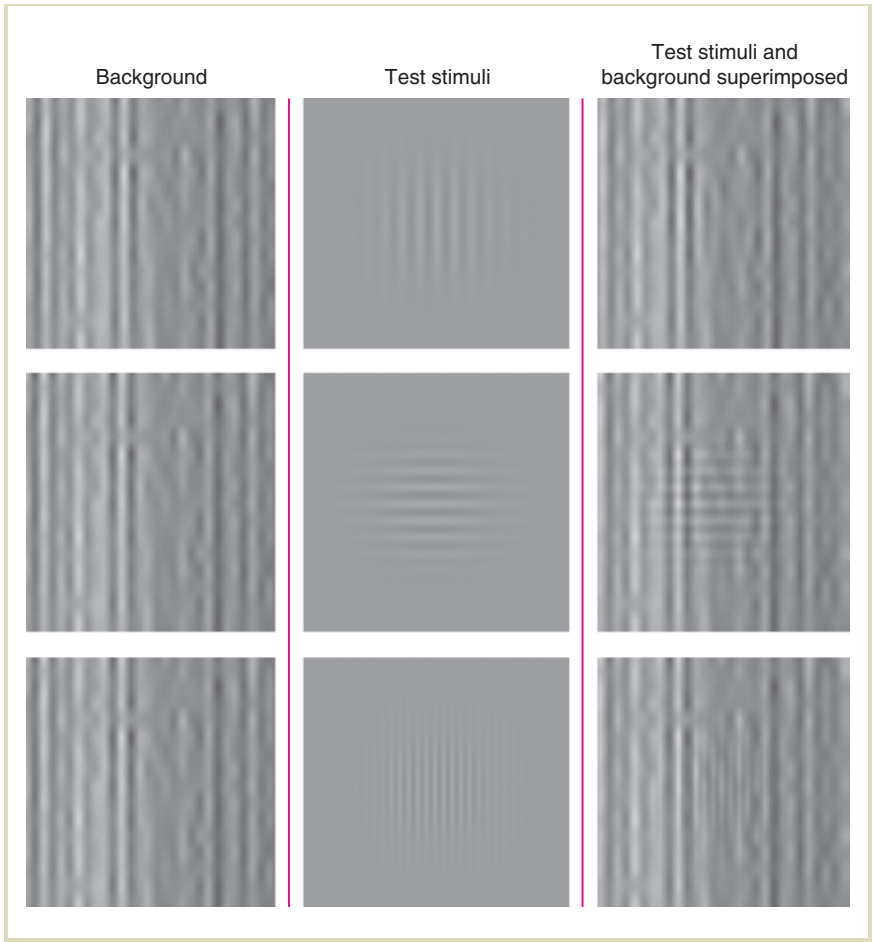
10.7.5 VISUAL MASKING AND PHASE UNCERTAINTY

The perceptual data behind the CSF have been derived in the so-called detection experiments, in which grating stimuli of various spatial frequencies have been imposed on the uniform background at a given adaptation luminance. Thus, the CSF (in fact its inverse as *threshold* $= 1/$*sensitivity*) specifies the *detection* contrast threshold, which is required to distinguish a given stimulus from the uniform background.

A similar experiment can be performed for nonuniform backgrounds, which actually contain other signals with various contrast, spatial frequency, and orientation characteristics. In this case, the grating stimuli should be discriminated from the cluttered background, and the resulting *discrimination* thresholds can be significantly different from the corresponding detection thresholds for the uniform background. This effect on discrimination thresholds by the background content is called "visual masking" or "spatial masking" and constitutes one of the most important characteristics of the HVS in complex image perception.

Figure 10.14 illustrates the visual masking effect. When the background (left column) and the test stimulus to be detected (central column) have similar spatial frequencies and orientations, the discriminability of the test pattern imposed on the background (right column) is reduced due to visual masking. In such conditions, the discrimination contrast threshold of the test stimulus is elevated with respect to the corresponding threshold in the CSF. The second row in Figure 10.14 shows the background and the test stimuli that are differently oriented, in which case visual masking is weaker. A similar observation holds when spatial frequencies of both interfering stimuli are significantly different as shown in the third row in Figure 10.14.

FIGURE 10.14 *Visual masking: The left column shows backgrounds. The middle column displays test stimuli, which are superimposed onto the background, as shown in the right column. The strongest masking effect can be observed when the test stimulus and background pattern feature similar spatial frequencies and orientations (top row). (Image courtesy of Scott Daly)*


Background
Test stimuli
Test stimuli and background superimposed

Legge and Foley [182] conducted a series of experiments to determine how the presence of one masking grating affects the discriminability of another test grating. Figure 10.15 depicts contrast discrimination threshold as measured for a sine-wave grating signal of 2.0 cpd imposed on masker gratings of varying spatial frequencies ranging from 1.0 to 4.0 cpd. The masker contrast has been increased from 0.05% to 50% of Michelson contrast. Note that maskers of weak contrast do not affect the discrimination threshold. For the masker contrast in the range of 0.15–1.5%, which is close to the detection threshold as measured by the CSF (Equation 10.17), the facilitation effect can be observed. In this case, the discrimination threshold actually drops below the detection threshold (marked by blue lines). For higher masker contrasts, the elevation of discrimination thresholds can be observed at all masking frequencies. The elevated part of each plot can be fitted to a straight line (in double logarithmic coordinates) with a slope in the range 0.52–0.71 [182].

The computational model of visual masking used in the HDR VDP is relatively simple. It is called the "threshold elevation model" [49]. Due to the relative independence of spatial masking on the spatial frequencies (Figure 10.15) and adaptation luminance, the same masking model can be applied to all channels [182], which requires that contrast signals are normalized into the detection threshold units. In the HDR VDP, this is the case due to the prior processing stages of JND-scaled luminance encoding and CSF modeling (see Figure 10.5). The threshold elevation model ignores any masking for the background (masker) pattern's contrast below 1 JND (the facilitation effect is relatively weak for complex images [49]) and elevates the discrimination thresholds for suprathreshold background contrast magnitudes. The contrast elevation factor $t_e^{k,l}[i, j]$ for the spatial channel k, orientation channel l, and pixel i, j can be computed as follows:

$$t_{\text{elevation}}^{k,l}[i, j] = \max\left[1.0, \left|c_n^{k,l}[i, j]\right|^s\right], \tag{10.20}$$

where $c_n^{k,l}[i, j]$ is the detection threshold-scaled contrast of the masking pattern as a result of the cortex transform. The parameter s controls the contrast elevation slope, and values ranging from 0.6 to 1.0 are usually assumed. Note that Daly in his VDP [49] used a more elaborate masking model that takes into account the learning

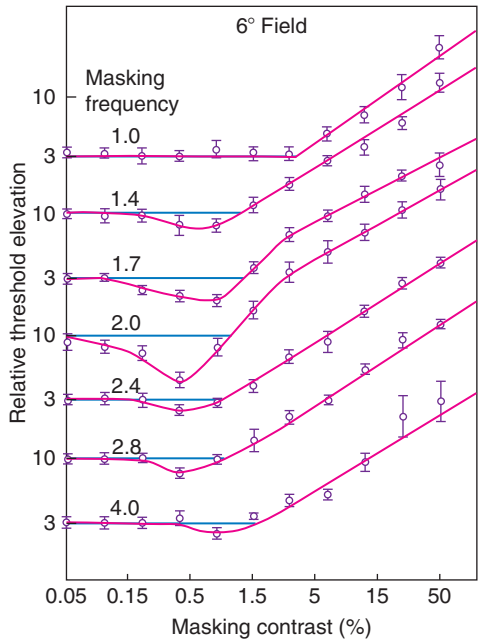


FIGURE 10.15 *Relative discrimination threshold changes due to spatial masking as a function of masking pattern contrast. These plots show the facilitation and threshold elevation effects as measured in the discrimination experiment for a sine-wave test grating of 2.0 cpd imposed on the background gratings of different spatial frequencies ranging from 1.0 to 4.0 cpd (the curves from top to bottom). The horizontal blue lines indicate unmasked detection threshold (as specified by the contrast sensitivity function) for the 2.0 cpd test grating. Each curve is plotted in its own scale. The test stimulus and maskers subtend a visual field of 6° × 6° (after [182]).*

effect, which effectively reduces visual masking for familiar (expected) signals and image patterns.

When the goal of HDR VDP processing is to detect the distortion visibility in the distorted image, the obvious candidate for the masking (background) pattern is the reference image. On the other hand, when the goal is just to compute visible differences between a pair of images, mutual masking can be considered by taking the lower value of masking for each pixel in the corresponding channels of the two images.

Figure 10.13 shows that the potential masking signal $c_n^{k,l}$ changes from negative to positive values. When such a masking signal is plugged directly as $|c_n^{k,l}[i, j]|$ into Equation 10.20, the predicted threshold elevation fluctuates accordingly, which effectively suggests an absence of masking at zero-crossing regions. However, experimental evidence demonstrates that the HVS is not sensitive to small (up to 90°) phase shifts, which are applied to the intrachannel signal [31]. This effect is called "phase uncertainty," and for this reason, low-pass filters with a support size roughly corresponding to the phase shift uncertainty for a given band are applied directly to the maps of contrast elevation factor $t_{\text{elevation}}^{k,l}$. Effectively, the phase uncertainty step in the HDR VDP results in more uniform spreading of the visual masking effect over textured regions, as would be expected [264,93].

The distortion signal $e_n^{k,l}$ is computed as the intrachannel difference between the JND-normalized contrast signals $c_{d/n}^{k,l}$ and $c_{r/n}^{k,l}$ in the distorted and reference images, respectively. To account for visual masking, the distortion signal is normalized by the contrast elevation factor $t_{\text{elevation}}^{k,l}$:

$$e_n^{k,l}[i, j] = \frac{\left| c_{r/n}^{k,l}[i, j] - c_{d/n}^{k,l}[i, j] \right|}{t_{\text{elevation}}^{k,l}[i, j]} \tag{10.21}$$

This normalization effectively reduces the magnitude of the distortion signal locally as the discrimination thresholds for the distortion are elevated due to the presence of masking patterns.

As evident from this discussion, visual masking is modeled in the HDR VDP independently for each channel, which is called "intrachannel masking." However, some masking effects can be observed between neighboring channels [96]. This so-called

interchannel masking [319,356] is ignored in the HDR VDP. The rationale for this choice is that visual masking is the strongest stimuli closely coupled in terms of their spatial location, spatial frequency, and orientation. To account for such coupling, the cortex transform performs the required decomposition of the reference and distorted images.

10.7.6 PSYCHOMETRIC FUNCTION AND PROBABILITY SUMMATION

The last two processing stages in the HDR VDP express the distortion visibility in terms of probabilities within each channel, and the probabilities are then summed across all channels. These steps are performed exactly in the same way as in the original VDP [49].

The psychometric function [233] describes the increase of detection probability $P^{k,l}[i, j]$ as the distortion contrast $e_n^{k,l}[i, j]$ increases:

$$P^{k,l}[i, j] = 1.0 - \exp^{-\left|\alpha \cdot e_n^{k,l}[i,j]\right|^h}, \tag{10.22}$$

where h is the slope of the function ($h = 3$), and $\alpha = [-\ln(1 - 0.75)]^{1/h}$ ensures that the probability $P^{k,l}[i, j] = 0.75$ for the contrast signal $e_n^{k,l}[i, j] = 1$ JND. The probability summation is performed using an equation based on product series:

$$P_t[i, j] = 1 - \prod_{k=1,\ldots,K;l=1,\ldots,L} \left(1 - P_m^{k,l}[i, j]\right), \tag{10.23}$$

where $P_t[i, j]$ denotes the total probability of distortion detection summed across all bands for a given pixel i, j.

The total probability is displayed as in-context difference maps as shown in Figure 10.16. In this example, an above-threshold sinusoidal grating is imposed upon a reference image. Note that the metric properly predicts that due to masking, the distortion is more visible in smooth image regions than in textured regions.

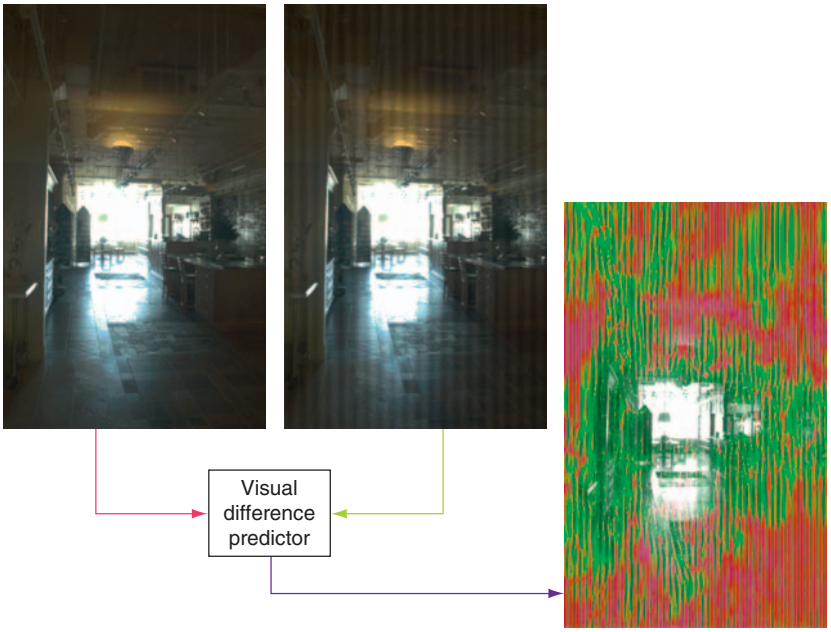


FIGURE 10.16 *HDR VDP processing example: The difference map between the reference image and the corresponding distorted image, which has a superimposed sine grating signal. The red color in the difference map denotes the probability of detecting distortions higher than 0.75, which roughly corresponds to 1 JND. (Image courtesy of Rafał Mantiuk; HDR source image courtesy of Byong Mok Oh)*

10.8 DYNAMIC RANGE–INDEPENDENT IMAGE-QUALITY METRIC

Typical perception-based image-quality metrics, including the HDR VDP discussed in the preceding section, commonly assume that the dynamic range of compared images is similar. Such metrics predict visible distortions using measures based on the magnitude of pixel intensity or normalized contrast differences between the two input images, which become meaningless when input images have significantly different contrast or luminance ranges.

However, when we look at images on a computer screen or even at traditional photographs, we often have an impression of plausible real-world depiction, although luminance and contrast ranges are far lower than in reality. This is also the case for HDR images, whose reproduction requires adjusting their dynamic range to match the capabilities of display devices using tone-mapping operators (TMOs) (see Chapter 8). The proliferation of new-generation display devices featuring higher dynamic range (Chapter 6) introduces the problem of enhancing legacy 8-bit images, which requires the use of so-called inverse tone-mapping operators (iTMOs), which are discussed in Chapter 9. An essential problem in these important applications is how to measure the effect of a dynamic range modification on the perceived image quality.

Perceived image quality can also be affected by the loss of important image features and other changes introduced to image structure. This aspect of image quality has been successfully addressed by the SSIM index (Section 10.5), but a deeper insight into the nature of such structural changes and precise determination of their visibility by the human eye would often be desirable. Also, the SSIM index is sensitive to local average luminance and contrast values, which makes it inadequate for a direct comparison of LDR and HDR images (see Figure 11 in [15]). In principle, the isolated structure comparison component $s(x, y)$ (Equation 10.9) of the SSIM index could potentially be used for this purpose, but this would require further investigation.

To address these problems, Aydın et al. [15] proposed the dynamic range–independent (DRI) image-quality metric, which performs perceptual scaling of detected changes in the image structure. Thus, the DRI metric can be considered a hybrid of contrast detection and SSIM metrics. The metric takes as input a pair

of images and includes an HVS model similar to that used in the HDR VDP, which enables precise detection of visible contrast changes. However, instead of reporting such changes immediately as the VDP, HDR-VDP, and VDM metrics, the visibility information is used to analyze only visible structural changes. Three classes of such structural changes are considered:

Loss of visible contrast: Image details that can be seen in the reference image disappear in the distorted image. This is a common problem in tone mapping resulting from strong contrast compression.

Amplification of invisible contrast: Image details that cannot be seen in the reference image are enhanced beyond the visibility threshold in the distorted image. This is a common problem for image-sharpening operations, which magnify the noise visibility. Also, TMOs, in particular those working in the gradient domain [89, 208] (see also Section 8.5.3), may enhance visibility of low-contrast details that cannot be seen in the HDR image. Finally, inverse tone mapping may introduce contouring (banding) artifacts due to excessive contrast stretching, which is applied to quantized 8-bit colors (see also Chapter 9).

Reversal of visible contrast: Image details can be seen in both input images, but their contrast has different polarity (signs). Such effects are usually associated with strong image manipulation such as clipping, low-bandwidth compression, and halo artifacts introduced by local TMOs.

All these distortions are considered at various scales and orientations that correspond to the visual channels as modeled by the cortex transform. Figure 10.17 illustrates each of the discussed distortion types for a simple signal that is a reasonable approximation of an intrachannel contrast signal as shown in Figure 10.13. By classifying intrachannel contrast in terms of its subthreshold or suprathreshold magnitude independently for both input images makes the DRI metric invariant to differences in dynamic range.

Figure 10.18 presents the processing flow in the DRI metric, which is identical to the HDR VDP up to the cortex transform (Figure 10.5). Then, JND-scaled contrast information for each spatial and orientation channel k, l is expressed in terms of detection probability $P_{\mathrm{v}}^{k,l}$ and its inverse $P_{\mathrm{i}}^{k,l} = 1 - P_{\mathrm{v}}^{k,l}$ using the psychometric function as specified in Equation 10.22. The detection probabilities are computed independently for the reference $P_{\mathrm{r/v|i}}^{k,l}$ and distorted $P_{\mathrm{d/v|i}}^{k,l}$ images. The

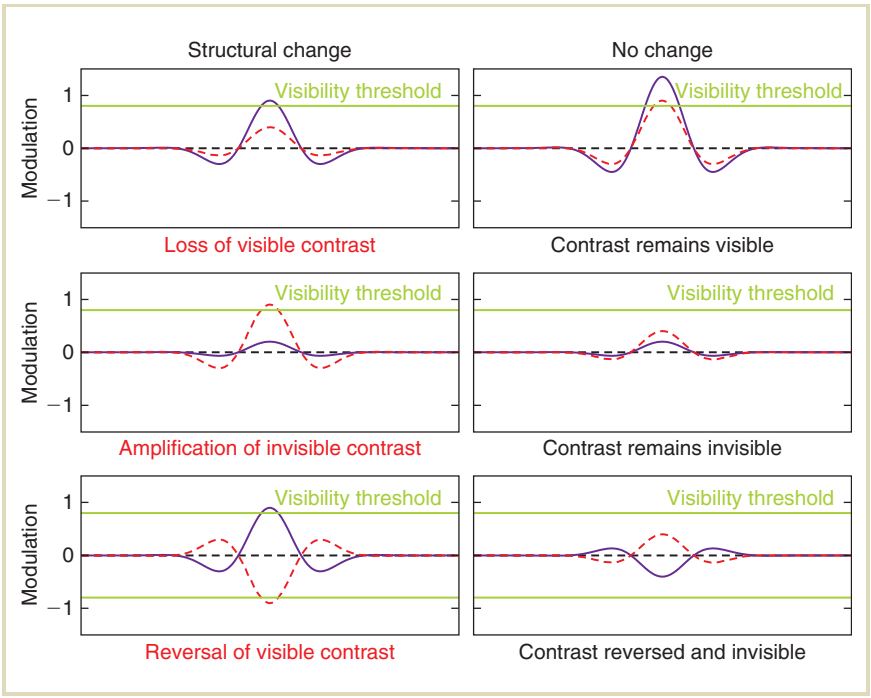


FIGURE 10.17 *Several cases of contrast modification that the DRI metric classifies as a structural change (left) or a lack of structural change (right). Purple continuous and pink dashed lines depict the reference and test (distorted) signals, while green horizontal lines denote the visibility threshold level.*

three structural distortion types are then computed as conditional probabilities:

$$\begin{aligned} &\text{loss of visible contrast:} && P_{\text{loss}}^{k,l} = P_{\text{r/v}}^{k,l} \cdot P_{\text{d/i}}^{k,l}, \\ &\text{amplification of invisible contrast:} && P_{\text{ampl}}^{k,l} = P_{\text{r/i}}^{k,l} \cdot P_{\text{d/v}}^{k,l}, \\ &\text{and reversal of visible contrast:} && P_{\text{rev}}^{k,l} = P_{\text{r/v}}^{k,l} \cdot P_{\text{d/v}}^{k,l} \cdot s^{k,l} \end{aligned} \tag{10.24}$$

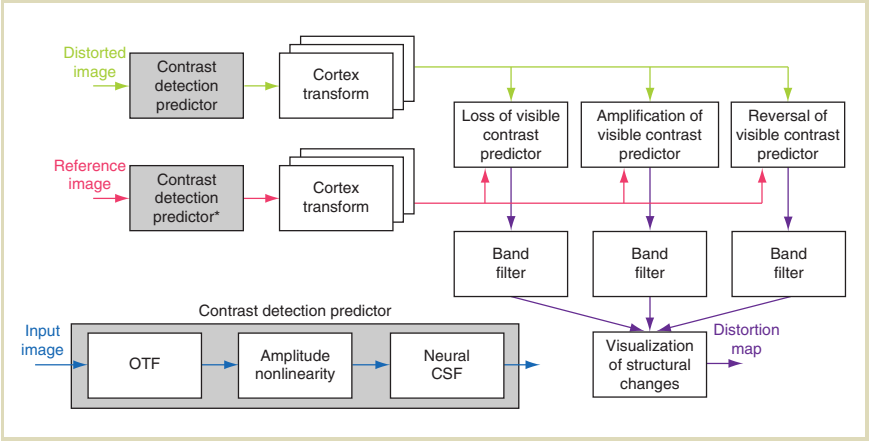


FIGURE 10.18 *The data flow diagram of the DRI metric. (Figure courtesy of Tunç Ozan Aydın and Rafał Mantiuk)*

where $s = 1$ if the polarity of contrast $c^{k,l}$ in the reference and distorted images differ:

$$s^{k,l} = \left[c_{\mathrm{r}}^{k,l} \cdot c_{\mathrm{d}}^{k,l} < 0 \right] \tag{10.25}$$

For simplicity, we omit the pixel indices $[i, j]$. Since in a nonlaboratory setup and for complex images the thresholds for contrast detection are higher, the psychometric function used to derive $P_{\mathrm{r/v}}^{k,l}$ and $P_{\mathrm{d/v}}^{k,l}$ is shifted toward higher contrasts so that the original 95% detection probability (2 JND units) corresponds to only 50% visibility probability in the shifted function (refer to the original paper for more details [15]).

The band filters, shown in Figure 10.18, remove signals that should not belong to a given channel but appeared in $P_{\mathrm{loss}}^{k,l}$, $P_{\mathrm{ampl}}^{k,l}$, and $P_{\mathrm{rev}}^{k,l}$ because of applying the nonlinear psychometric function to contrast $c^{k,l}$ resulting from the cortex transform. Essentially, the cortex transform filters are applied for the second time to remove all

frequencies that do not belong to a given channel. An analogy to the standard analysis and synthesis steps in wavelet processing can be found here [186] (see also Section 8.3.5). The probability summation as specified in Equation 10.23 is performed over all channels k, l independently for filtered probability maps $\hat{P}^{k,l}_{\text{loss}}$, $\hat{P}^{k,l}_{\text{ampl}}$, and $\hat{P}^{k,l}_{\text{rev}}$. As a result, three maps with summed probabilities for each distortion type are obtained.

In the following sections, we present selected applications of the DRI metric, and we then compare its performance with respect to SSIM and HDR VDP.

10.8.1 APPLICATIONS

While the DRI metric finds applications in many highly specialized tasks, a simple comparison of images with significantly different contrast and brightness is often needed. Figure 10.19 demonstrates the DRI metric outcome for various HDR-HDR, HDR-LDR, LDR-HDR, and LDR-LDR image combinations. The LDR image has been obtained through a simple tone mapping of the HDR image. In addition, in each compared image pair, random noise with increasing amplitude toward the distorted region center is locally added to the distorted image. This is the only image difference between the HDR-HDR and LDR-LDR image pairs (refer to the two top rows in Figure 10.19), for which visible contrast reversal (marked in red) and amplification of invisible contrast (blue) are reported by the metric.

For the image pairs HDR-LDR and LDR-HDR, similar noise predictions in the noisy image region hold, but more differences are additionally reported due to distortions introduced by tone mapping. When the HDR image serves as the reference image, loss of visible contrast (green) is reported in many regions in the LDR image (third row). On the other hand, when the LDR image serves as the reference, visible details in the HDR image are interpreted this time as amplification of invisible contrast.

More specialized applications of the DRI metric include the evaluation of TMOs and their inverse counterpart (iTMO), where contrast is actually expanded from LDR to HDR. Also, the analysis of image appearance on displays with various characteristics becomes possible using this tool [15]. For example, Figure 10.20 shows the comparison of an HDR image with corresponding tone-mapped images. The luminance ranges of $[0.24-89300]$ and $[0.1-80]\,\text{cd/m}^2$ have been assumed for the original scene and displayed tone-mapped image, respectively.

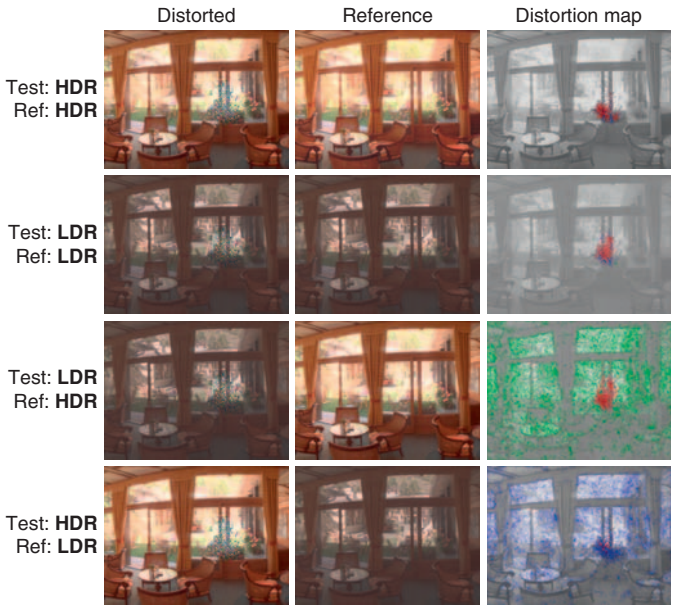


FIGURE 10.19 *Comparing images with different dynamic ranges. While distortions caused by the locally added noise are visible in all results, in the LDR-HDR (third row) and HDR-LDR (fourth row) cases, additional visible contrast loss and invisible contrast amplification can be observed, respectively. This is an expected effect due to contrast compression in tone mapping. HDR images are tone mapped using Reinhard's photographic tone reproduction for printing purposes. (Images courtesy of Tunç Ozan Aydın; original HDR photograph courtesy of SPHERON-VR © 2008 ACM, Inc. Reprinted by permission.)*

Three TMOs (one global and two local operators) have been considered: Drago's adaptive logarithmic mapping [71], Fattal's gradient domain compression [89], and Reinhard's photographic tone reproduction [274] (refer to Section 8). For these TMOs, certain loss of detail can be observed in the outdoor region due to

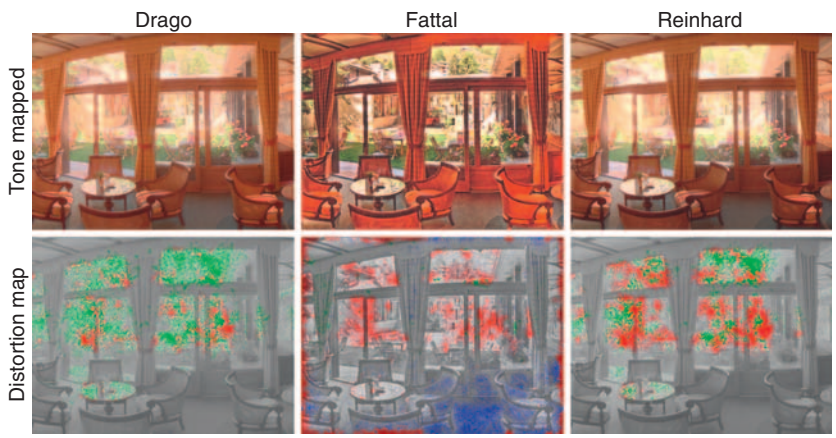


FIGURE 10.20 *Comparison of tone-mapping operators. (Images courtesy of Tunç Ozan Aydın; original HDR photograph courtesy of SPHERON-VR)*

strong contrast compression. Pixel intensity saturation (clipping) also causes visible contrast reversal, which is reported for some pixels as the dominant (strongest) distortion.

Drago's operator reproduces contrast relatively well in dark image regions and tends to wash out image details in light regions due to the logarithmic shape of the tone-mapping curve. Reinhard's operator leads to a better detail reproduction due to local dodging and burning mechanisms. The detail amplification typical for Fattal's operator can be seen in darker scene regions, which in real-world observation conditions are more difficult to see due to insufficient HVS sensitivity. The results as predicted by the DRI metric are in good agreement with the expected outcome of each particular TMO, which suggests the potential use of the DRI metric as a diagnostic tool for such algorithms.

10.8.2 COMPARISON WITH SSIM AND HDR VDP

Figure 10.21 shows a side-by-side comparison of the SSIM, HDR VDP, and DRI quality metrics, where a blurred and a sharpened version of the reference is used as the distortion image. The metric predictions are quite similar. However, the HDR

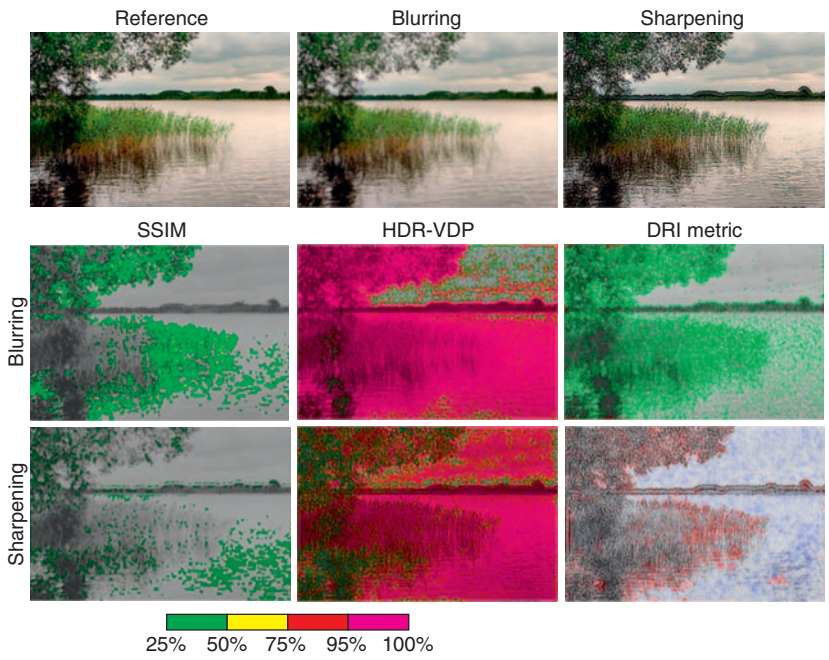


FIGURE 10.21 *The reference, blurred, and sharpened test images (top row) and DRI metric responses to blurring (middle row) and sharpening (bottom row). Color coding for SSIM and HDR VDP is given in the scale. Color coding for the dynamic range–independent metric is the same as in Figure 10.19. (Images courtesy of Tunç Ozan Aydın © 2008 ACM, Inc. Reprinted by permission.)*

VDP responds stronger than the DRI metric, as all visible differences are reported (not only those leading to structure changes).

The SSIM's response is the weakest, possibly because of difficulties in calibration for distortions with magnitude near the visibility threshold. The important difference between the DRI metric and the others is the classification of distortion types. That is, in the case of blurring, the DRI metric classifies all distortions as loss of visible contrast, confirming the fact that high-frequency details are lost. On the other hand, in the case of sharpening, we observe contrast reversal and amplification of invisible contrast, both of which are expected effects of unsharp masking. This classification gives insight into the nature of the image-processing algorithm and would enable distortion-type-specific further processing.

10.9 SUPRATHRESHOLD HDR IMAGE-QUALITY METRICS

As discussed in Sections 10.7 and 10.8, the HDR VDP and DRI metrics are typical threshold metrics that enable us to precisely judge the visibility of differences in a distorted image with respect to a reference image. Essentially, the HDR VDP classifies distortions into visible (suprathreshold) and invisible (subthreshold), while the DRI metric detects the presence of structural changes explicitly near the threshold contrast values.

However, while the threshold metrics are focused on the task of visible distortion discrimination or structural change detection, they are not suitable to estimate the magnitude of such distortions or structural changes. Even if these metrics normalize the physical contrast value c into the JND unit using the contrast sensitivity S derived from the advanced CSF model (Equation 10.17), the problem is that simple linear scaling $S \cdot c$ is used for this purpose. In the case of HDR VDP, the JND-scaled contrast is also normalized (Equation 10.21) to account for threshold elevation due to masking (Equation 10.20), but this normalization is correct only if the contrast signals in the distorted and reference images have similar magnitudes. For suprathreshold distortions, the discrimination thresholds (as shown in Figure 10.15) can be quite different, which means that each of the contrast signals in the denominator of Equation 10.21 should be normalized differently.

More advanced models of visual masking that are suitable for handling suprathreshold contrast differences include transducer functions [182,362,194,208].

Essentially, transducer functions measure the human visual response R (i.e., apparent contrast) to the physical contrast c.

The idea of transducer derivation is based on Fechner's law, in which sensory response is expressed as an integral of discrimination thresholds. Mantiuk et al. [207] follow Fechner's procedure to derive the JND-scaled space (see Sections 4.1.1 and 10.7.2), which models the HVS response to the full range of visible luminance that is adequate for HDRI applications. As opposed to *luminance increments* on a uniform background, which led to Fechner's law and its HDR extension, in this section, we are concerned with *contrast increments*. The transducers discussed in the following sections are the result of Fechner-like transducer derivation but are applied this time to contrast discrimination data [182,96,359] in the presence of background contrast (similar to that shown in Figure 10.15). For a detailed description of the contrast transducer derivation, refer to [362,208].

Another way of deriving the human visual response R to contrast is to measure it directly in magnitude-estimation experiments, where the perceived contrast is numerically estimated by human observers for randomly presented stimuli of wide physical contrast range. Note that the magnitude estimation method has been successfully used by Stevens and Stevens [305] to find the relation between luminance L and perceived brightness B, which resulted in the well-known Stevens' power law $B = k_1 L^a$. Since the power law better predicts brightness sensation than Fechner's law $B = k_2 \ln L$, one can expect that magnitude-estimation experiments could lead to a more precise perceived contrast model than transducer functions, which are based on the data acquired in increment-threshold experiments [182,96].

Cannon used contrast magnitude-estimation data to approximate the relation between perceived contrast P_c and physical contrast c [33]:

$$P_c = k(c - \Delta c)^{\alpha} \tag{10.26}$$

Here, Δc denotes the perceived contrast threshold for given adaptation conditions, which can be derived from the CSF given in Equation 10.17. The value of α is in the range [0.40, 0.52] and depends on spatial frequency of contrast stimuli, its spatial extent, and eccentricity.

Cannon shows that the perceived contrast measure P_c is in good agreement with the Legge–Foley transducer function [182] for stronger contrasts but does not predict the facilitation effect for near-threshold contrast values. In fact, in this particular

range, the magnitude-estimation data do not provide enough measurement points, which may cause some inaccuracies.

Our further discussion is focused on transducers, which are more commonly used in image-quality metrics. In the following section, we describe Wilson's transducer, which is used in many image-quality metrics and contrast-processing applications. In Section 10.9.2, we present Mantiuk's transducer, which has been derived specifically for a wide range of contrasts, as typically seen in HDR images. Finally, we present an example of Wilson's transducer applied to a suprathreshold quality metric that is suitable for HDR images.

10.9.1 WILSON'S TRANSDUCER

As an example of the transducer function R, which has been used in the context of HDR applications [202,17], we present the formulation proposed by Wilson [362]:

$$R(S \cdot c) = \frac{3.291 \cdot \left\{ \left[1 + (S \cdot c)^3 \right]^{\frac{1}{3}} - 1 \right\}}{0.2599 \cdot [3.433 + S \cdot c]^{0.8}}, \tag{10.27}$$

where c is a measure of contrast and S stands for sensitivity as specified in Equation 10.17. The remaining coefficient values result from fitting this function to experimental contrast-discrimination data (including those shown in Figure 10.15) and the assumption that if $S \cdot c = 1$ threshold detection unit, then $R(1) = 1$.

Figure 10.22 shows the response R for two different luminance adaptation values L_a and spatial frequencies ρ of the contrast signal. For $L_a = 400\,\mathrm{cd/m^2}$ and $\rho = 5\,\mathrm{cpd}$, the corresponding sensitivity $S = 190$ (as derived from Equation 10.17) is much higher than $S = 45$ for $L_a = 1\,\mathrm{cd/m^2}$ and $\rho = 1.5\,\mathrm{cpd}$. Such higher sensitivity S suggests a stronger response of the HVS to contrast, which results in the steeper plot of R.

As can be seen in the inset, which shows the same plots but for a much wider contrast range, at high-contrast values, the response is compressed, while the curve is much steeper for low-contrast values near the threshold. In the latter case, the facilitation effect is modeled, when the contrast-discrimination thresholds fall below the detection thresholds as measured for the uniform (masker-free) background.

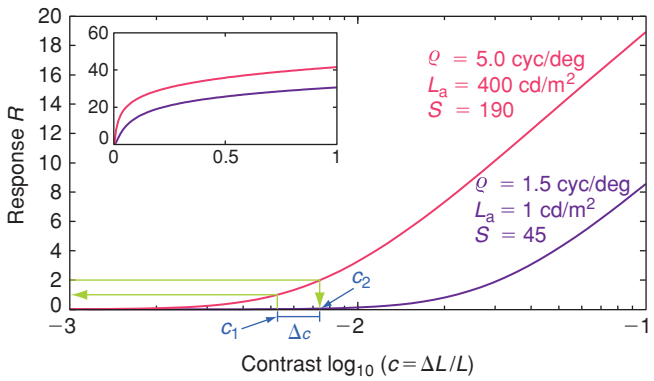


FIGURE 10.22 *Wilson's transducer: The HVS response R to contrast c for two different luminance adaptation values L_a and spatial frequencies ρ of the contrast signal (refer to the text for a more detailed discussion). (Plot courtesy of Rafał Mantiuk)*

Note also that by construction, for any pair of values, the background contrast c_1 and the corresponding contrast-discrimination threshold Δc (such a pair can be derived from Figure 10.15), the following relation holds [362]:

$$R(S \cdot (c_1 + \Delta c)) - R(S \cdot c_1) = 1 \tag{10.28}$$

Thus, for a given contrast c_1, it is possible to determine from the transducer plot the corresponding response R. Then, by adding 1 JND, the response $R+1$ is obtained, and by referring again to the transducer plot, one can find the corresponding contrast c_2. This is illustrated in Figure 10.22 with green arrows: Follow the arrow from c_1 to $R=1$ and then from $R=2$ to c_2. The difference $\Delta c = c_2 - c_1$ directly corresponds to the contrast-discrimination threshold for the background contrast c_1.

Similar transducer functions are embedded in multichannel image-quality metrics such as VDM [194], in which case contrast $c^{k,l}$ is measured for each spatial

frequency k and orientation channel l, and only intrachannel masking is considered (similar to the HDR VDP). The use of the transducer function and subsequent error pooling across all channels k, l and all pixels i, j using the Minkowski summation results in estimation of the image difference magnitude D directly scaled in JND units:

$$D = \left(\sum \left| R_r^{k,l}[i,j] - R_d^{k,l}[i,j] \right|^{\beta} \right)^{\frac{1}{\beta}}, \tag{10.29}$$

where $R_r^{k,l}$ and $R_d^{k,l}$ stand for the intrachannel transducer responses to contrast in the reference and distorted images.

In the following section, we discuss a successful attempt of deriving a specialized transducer function suitable for extremely wide contrast ranges.

10.9.2 MANTIUK'S TRANSDUCER

As noticed by Mantiuk et al. [208], contrast-discrimination data is normally obtained for relatively limited contrast levels of up to $M = 50\%$, measured as Michelson contrast [182,96]. This measure of contrast is given by

$$M = \frac{|L_{\max} - L_{\min}|}{L_{\max} + L_{\min}}, \tag{10.30}$$

This is inadequate for contrast values in HDR applications. Much wider contrast ranges of $M \approx 99.5\%$ have been considered in psychophysical experiments conducted by Whittle [359]. Mantiuk et al. used a similar procedure as in [362] using Whittle's data to derive a simple transducer function $R(G)$:

$$R(G) = 54.09288 \cdot G^{0.4185}, \tag{10.31}$$

where G is the logarithmic ratio contrast measure $G = \log_{10} \frac{L_{\max}}{L_{\min}}$ and $L_{\min}$ and $L_{\max}$ denote luminance values for a pair of pixels. The contrast measure G is particularly convenient in image-processing applications, in which each pixel luminance L is converted into a more perceptually uniform brightness measure $I[i,j] = \log_{10} L[i,j]$ to approximate the visual response to luminance. Then, by computing a simple difference such as $G[i,j] = |I[i,j] - I[i,j+1]|$, the logarithmic ratio contrast measure can be found between a pair of neighboring pixels.

Note that contrast G is linked with commonly used Michelson's contrast M and Weber's contrast W:

$$W = \frac{L_{\max} - L_{\min}}{L_{\min}} \tag{10.32}$$

by simple relations:

$$G = \log_{10}\left(\frac{1+M}{1-M}\right) \tag{10.33a}$$

$$G = \log_{10}(W+1) \tag{10.33b}$$

The transducer proposed by Mantiuk is convenient in approximating the human visual response to a wide range of contrasts, directly scaled in JND units. Due to the simplicity of the transducer, the corresponding inverse function is straightforward to obtain:

$$G(R) = 7.2232 \cdot 10^{-5} \cdot R^{2.3895} \tag{10.34}$$

This is especially important when contrast modification is performed over the intuitive and perceptually scaled R, which much better approximates perceived contrast than the physical contrast measure G. Then, Equation 10.34 can be used to derive G from R and finally reconstruct luminance L in the modified image. In Section 8.5.4, we present an example of a perceptual framework for contrast processing, which implements this idea in the context of tone mapping [208].

A disadvantage of Mantiuk's transducer is that it does not account for luminance adaptation values, which effectively means that the lower contrast sensitivity of the HVS in dark (scotopic) conditions is not properly modeled. The fit of Equation 10.31 to Whittle's measurement data is worse near the contrast threshold. For this reason, a more complex approximation that improves this fit is given by Mantiuk et al. [208]. Further, it is not clear how to account for more advanced effects such as maladaptation (see Section 10.10). In this respect, Wilson's transducer can be endorsed, and in the following section, we present its successful application in the context of HDR imaging.

10.9.3 A METRIC BASED ON WILSON'S TRANSDUCER

Mantiuk et al. [202] have built a custom quality metric suitable for estimating the magnitude of suprathreshold distortions in HDR images based on Wilson's transducer.

First, the logarithm of image luminance values $I = \log_{10} L$ is computed to approximate the human visual response to luminance. Then, each image is decomposed into spatial frequency channels using the Laplacian pyramid [28] (orientation channels have been ignored for performance reasons).

Note that in Laplacian pyramids, the band-pass signal in each channel k is computed as the difference $G_k = I_k - I_{k+1}$, where I_k and I_{k+1} denote the input image I convolved with Gaussian filters of increasing size (spatial support). Thus, the value of $G_k[i, j]$ is a logarithmic ratio contrast measure, as introduced in Section 10.9.2, which can be converted into the familiar Weber contrast W using the relation $W = 10^{|G|} - 1$ (Equation 10.33). The resulting contrast W can be normalized into detection threshold units $S \cdot W$ using the CSF function (Equation 10.17). Then, the normalized contrast can be directly plugged into Wilson's transducer (Equation 10.27). The JND-scaled difference magnitude can be measured between corresponding channels for a pair of input images, or the Minkowski summation (Equation 10.29) can be used to compute the overall difference.

Note that through the CSF equation, the transducer dependency on adaptation luminance and spatial frequency can be modeled. Since Equation 10.17 is suitable for HDR luminance ranges, this is also the case for the sensitivity S, and thus contrast W is properly normalized to JND units. While Wilson's transducer has not been explicitly fit to discrimination data for very high contrasts, its compressive behavior in such contrast range seems to be plausible (see Figure 10.22). This way, by combining carefully selected HVS models, which are suitable for luminance and contrast ranges typical for HDR applications, a viable solution for suprathreshold image differences has been proposed.

Mantiuk et al. [202] also demonstrate an application of this metric to measure the contrast distortions in reproduced images for a given display device and ambient lighting conditions. The contrast distortions are measured with respect to the appearance of the original image as it would be seen under real-world observation conditions. An optimization process is performed over the shape of the tone-mapping curve to minimize the distortions by taking into account not

only the display characteristics but also the content of displayed images and video. Section 8.3.6 presents this tone-reproduction technique in more detail.

10.10 ACCOUNTING FOR PARTIAL ADAPTATION

So far, it has been tacitly assumed that the eye is perfectly adapted to the actual viewing conditions. For instance, the HDR VDP and DRI metrics make the unrealistic assumption that full adaptation to each pixel's luminance occurs. From the standpoint of distortion estimation, this is a conservative approach, which accounts for the worst-case scenario of perfect visibility.

In practice, the visibility of distortion is usually reduced due to the maladapted state of the HVS. In such conditions, both contrast-detection and contrast-discrimination thresholds (Section 10.7.5) may increase when the adaptation luminance L_a is different from the actual (background) luminance L. In many practical applications, it is useful to model such reduced visual sensitivity. For example, in the task of estimating critical information visibility, it is important to ensure that a given text is readable even if the observer is maladapted [17].

A state of maladaptation is typical in real-world observation conditions. For instance, saccades or intentional gaze changes may cause the eye to focus on a region with an average luminance that is significantly different from the current adaptation luminance. Also, lighting conditions may change significantly, for instance when driving a car, even when watching modern HDR displays.

There are implications of maladaptation in terms of visible contrast thresholds. Figure 10.23 shows in pink the standard CVI curve (see also Figure 10.7), which relates the contrast-detection threshold to background luminance, assuming that the HVS is fully adapted, that is, $L_a = L$. The purple curves depict the threshold contrast for three different adaptation states L_a equal to 1100 and 10 000 cd/m^2. The pink curve can be thought as an envelope of all purple curves for continuously changing L_a values. Note that each purple curve intersects the pink curve exactly at one point when $L_a = L$, that is, in the steady state of perfect adaptation to L. Figure 10.23 also shows how the threshold contrast increases (ΔC_1 and $\Delta C_{10,000}$) when the eye is adapted to $L_a = 1$ and $L_a = 10{,}000\,\text{cd/m}^2$, while the background luminance $L = 100\,\text{cd/m}^2$ is observed.

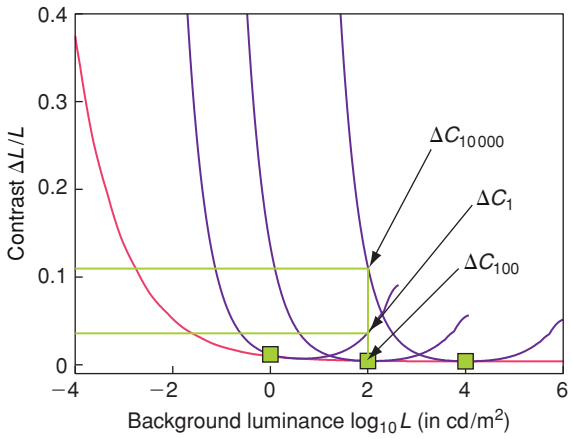


FIGURE 10.23 *The CVI function (pink curve): Detection threshold contrast as a function of background luminance L assuming perfect adaptation $L_a = L$. Purple curves: Three instances of the CVIA function for adaptation luminances L_a of 1, 100, and 10 000 cd/m^2. At $L = 100$ cd/m^2, threshold contrast (ΔC) is the lowest for $L_a = 100$ cd/m^2, whereas for maladapted states $L_a = 1$ and $L_a = 10\,000$ cd/m^2, the threshold increases notably. (Plot courtesy of Tunç Ozan Aydın)*

The model of maladaptation as presented in Figure 10.23 closely follows the CVI and adaptation CVIA(L, L_a) functions proposed by Irawan et al. [135]. The CVIA(L, L_a) function specifies just-noticeable contrast differences when the eye is looking at luminance level L, while it is fully adapted to luminance level L_a. The CVIA function is derived from the photoreceptor response (see Section 8.1.1) and calibrated so that it matches the standard CVI function when $L_a = L$, that is, CVIA(L_a, L_a) = CVI(L_a).

Aydın et al. [17] extended the suprathreshold HDR quality metric presented in Section 10.9.3 to account for maladaptation using the CVIA(L, L_a) function.

While computing the sensitivity S using the CSF from Equation 10.17, they performed the following normalization:

$$S = \mathrm{CSF}(\rho, L_\mathrm{a}, \ldots) \cdot \frac{\mathrm{CVI}(L_\mathrm{a})}{\mathrm{CVIA}(L, L_\mathrm{a})}. \tag{10.35}$$

Note that the full list of CSF parameters is explained in Section 10.7.3. By plugging the sensitivity S into Wilson's transducer (Equation 10.27), the reduction of response R to contrast in the state of maladaptation can be modeled, as illustrated in Figure 10.24.

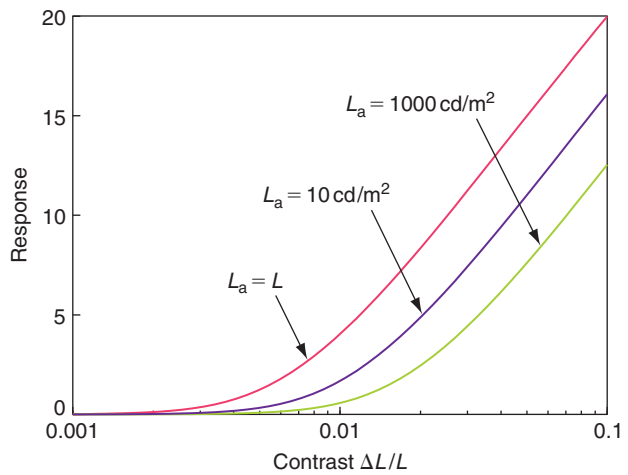


FIGURE 10.24 *Wilson's transducer response for background luminance* $L = 100\,\mathrm{cd/m^2}$, *at adaptation levels* L_a: *100 (pink), 10 (purple), and 1000 (green)* $\mathrm{cd/m^2}$ *for stimuli of spatial frequency of 4 cpd. (Plot courtesy of Tunç Ozan Aydın)*

Pattanaik et al. [248] and Irawan et al. [135] discuss temporal aspects of converging L_a to L by modeling neural and photochemical mechanisms of luminance adaptation in the cone and rod systems in the retina of the eye. Aydın et al. [17] followed these techniques to model dynamic changes in Wilson's transducer response for HDR video sequences by computing L_a based on the history of pixel intensities in previous frames. The value of L is locally derived from the pixel luminance in the current frame. More details on temporal adaptation in the context of tone mapping for video sequences are discussed in Section 7.5.

Aydın et al. [17] also extended the DRI metric to account for maladaptation. They achieved this effect by using the $\mathrm{CVIA}(L, L_a)$ function instead of the standard $\mathrm{CVI}(L_a)$ function in the derivation of JND-scaled space (Section 10.7.2). In addition, they show an application of the extended DRI and suprathreshold metrics to analyze the readability of information displayed on LCD panels under temporally and spatially changing external lighting conditions.

10.11 SUMMARY

In this chapter, we discuss the problem of image-quality evaluation for HDR images. Taking into account the state of research on this topic, we focused on metrics that require full access to the reference image to measure distortions in the test (distorted) image. While the use of standard pixel-based distortion metrics such as MSE and PSNR is possible for this purpose, we argue that to obtain more reliable quality predictions, it is highly recommended to transform HDR pixel intensities into a more perceptually uniform space that correlates well with perceived lightness. Under this condition, the MSE and PSNR might give reliable distortion predictions, particularly in such applications as image compression, which actively use human visual models to distribute the distortions in a perceptually uniform way over the whole image.

For distortions of a more local nature as introduced by many image-processing operations, much better predictions of image quality can be obtained using, for example, the SSIM index or suprathreshold metrics. These metrics routinely measure the distortion magnitude and take into account the spatial pattern of neighboring pixel intensities, which enables the capture of structural changes and better account for local luminance adaptation. For small near-threshold distortions that should

be precisely classified as visible or invisible, we endorse perception-based metrics such as the HDR VDP. Finally, for the task of comparing images with significantly different dynamic ranges, the DRI metric can be used, which additionally classifies certain distortion types.

We hope that by reading this chapter, some intuition can be built regarding which metric is suitable for a given task. Even deeper insight can be obtained by experimenting directly with the discussed metrics. For example, the DRI and HDR VDP metrics are available as a free online service:

`http://drim.mpi-inf.mpg.de/generator.php`,

where a pair of images can be submitted and the resulting difference map is generated. A more advanced reader is encouraged to actively experiment with source code of perception-based metrics as the whole or selected HVS models, which can be useful in many other applications dealing with image perception modeling. Rafał Mantiuk made available (under the GPL license) the source code of the HDR VDP, which is accompanied with detailed documentation:

`http://hdrvdp.sourceforge.net`.

The suprathreshold metric discussed in Section 10.9.3 is embedded into the tone mapping pfstmo_mantiuk08, which is a part of *pfstmo* and *pfstools*:

`http://pfstools.sourceforge.net/`.

The problem of HDR image-quality estimation is still in its infancy. Existing metrics still cannot perform many practical quality-evaluation tasks in a reliable way. For example, the DRI metric counterpart that can measure the magnitude of differences between an image pair of significantly different dynamic range and meaningfully account for perceived quality loss has not been developed so far.

More research is also needed to account for maladaptation and temporal aspects of HDR video quality. In particular, accounting for maladaptation might be required for video sequences reproduced on HDR displays and projectors. Also, NR metrics that can meaningfully account for image quality need to be developed. While such metrics are developed for LDR images [35,368,339], they should be adapted to

handle HDR images. The issue of color handling definitely requires further investigation, and while iCAM06 [168] models many important color appearance phenomena, the issues of spatial vision are treated in a somewhat simplistic manner. An advanced visual model that takes into account visual adaptation, suprathreshold contrast, and spatial and color vision has been proposed in [246] with tone mapping as the main application (see Section 8.2.1). However, the model is general enough to be considered in other applications such as image-quality evaluation, optimizing image compression, or steering realistic image synthesis.

This page intentionally left blank

Image-Based Lighting

11

The earlier chapters in this book have described numerous properties and advantages of high dynamic range (HDR) imagery. A major advantage is that HDR pixel values can cover the full range of light in a scene and can be stored as calibrated linear-response measurements of incident illumination. Earlier chapters have described how these images are useful for improved image processing, and for determining how a human observer might perceive a real-world scene, even if shown on a low dynamic range (LDR) display.

This chapter describes how HDR images can be used as sources of illumination for computer-generated objects and scenes. Because HDR images record the full range of light arriving at a point in space, they contain information about the shape, color, and intensity of direct light sources, as well as the color and distribution of the indirect light from surfaces in the rest of the scene. Using suitable rendering algorithms, we can use HDR images to accurately simulate how objects and environments would look if they were illuminated by light from the real world. This process of using images as light sources is called "image-based lighting" (IBL). In that IBL generally involves the use of HDR images, both the IBL process and the HDR images used for IBL are sometimes referred to as HDRI, for high dynamic range imagery.

Figure 11.1 compares a simple scene illuminated by a traditional computer graphics light source (a) to its appearance as illuminated by three different IBL environments ([b] through [d]). The geometry of the scene consists of simple shapes and materials such as plastic, metal, and glass. In all of these images, the

© 2010, Elsevier Inc. All rights reserved.
DOI: 10.1016/B978-0-12-374914-7.00011-7

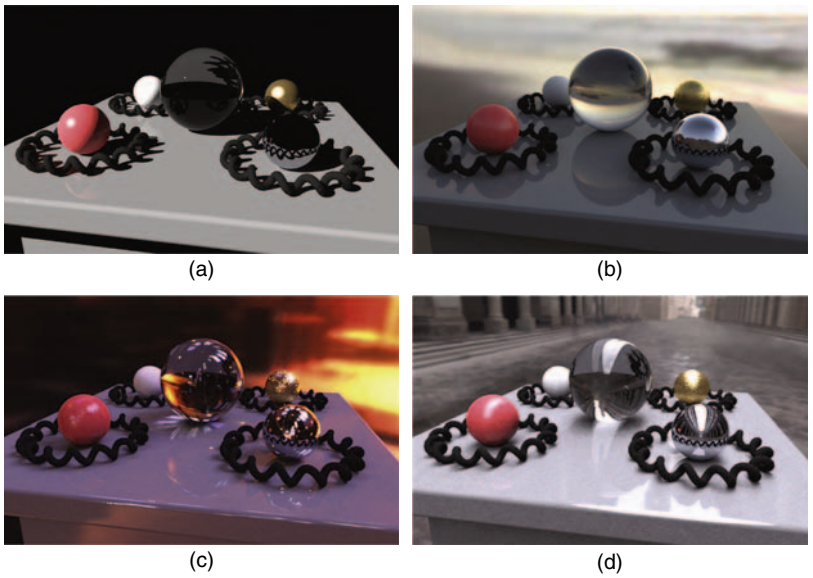


FIGURE 11.1 *Scene illuminated with (a) a traditional point light source, (b–d) HDR image-based lighting (IBL) environments, including (b) sunset on a beach, (c) inside a cathedral with stained-glass windows, and (d) outside on a cloudy day.*

lighting is being simulated using the RADIANCE global illumination system [349]. Without IBL (Figure 11.1[a]), the illumination is harsh and simplistic, and the scene appears noticeably computer-generated. With IBL (Figures 11.1[b] through 11.1[d]), the level of realism and visual interest of the scene are increased—the shadows, reflections, and shading all exhibit complexities and subtleties that are realistic and internally consistent. In each rendering, a view of the captured environment appears in the background behind the objects. As another benefit of IBL, the objects appear to actually belong within the scenes that are lighting them.

In addition to HDR photography, IBL leverages two other important processes. One of them is omnidirectional photography, the process of capturing images that see in all directions from a particular point in space. HDR images used for IBL generally need to be omnidirectional because light coming from every direction typically contributes to the appearance of real-world objects. This chapter describes some common methods of acquiring omnidirectional HDR images, known as "light probe image" or "HDR environment maps," which can be used as HDR IBL environments.

The other key technology for IBL is *global illumination*: rendering algorithms that simulate how light travels from light sources, reflects between surfaces, and produces the appearance of the computer-generated objects in renderings. Global illumination algorithms simulate the interreflection of light between diffuse surfaces, known as "radiosity" [108], and can more generally be built on the machinery of *ray tracing* [360] to simulate light transport within general scenes according to the *rendering equation* [145]. This chapter not only describes how such algorithms operate but also demonstrates how they can be used to illuminate computer-generated scenes and objects with light captured in light probe images.

An important application of IBL is in the area of motion-picture visual effects, where a common effect is to add computer-generated objects, creatures, and actors into filmed imagery as if they were really there when the scene was photographed. A key part of this problem is to match the light on the computer-generated (CG) elements to be plausibly consistent with the light present within the environment. With IBL techniques, the real illumination can be captured at the location at which the CG object needs to be placed, and then used to light the CG element so that it has the same shading, shadows, and highlights as if it were really in the scene. Using this lighting as a starting point, visual effects artists can augment and sculpt the IBL to achieve effects that are both dramatic and realistic.

Within the scope of IBL, there are several variants and extensions that increase the range of application of the technique. When implemented naïvely in global illumination software, IBL renderings can be computationally expensive for scenes with concentrated light sources. This chapter presents both user-guided and automatic techniques for *importance sampling* that make IBL calculations more efficient. In addition, various approximations of IBL can produce convincing results for many materials and environments, especially under appropriate artistic guidance. One is *environment mapping*, a precursor to IBL that yields extremely efficient and often

convincing renderings by directly mapping images of an environment onto object surfaces. Another technique, *ambient occlusion*, uses some of the machinery of IBL to approximate the self-shadowing of an object so that it can be quickly applied to different lighting environments. To begin, we will start with a detailed example of IBL in a relatively basic form.

11.1 BASIC IBL

This section describes IBL in both theoretical and practical terms using the example of *rendering with natural light* (RNL), an IBL animation shown at the SIGGRAPH 98 Electronic Theater. The RNL scene is a still life of computer-generated spheres on a pedestal, and is illuminated by light captured in the Eucalyptus grove at the University of California at Berkeley. The animation was modeled, rendered, and illuminated using the RADIANCE lighting simulation system [349], and the necessary scene files and images for creating the animation are included on the companion DVD-ROM. The animation was created via the following steps:

1. Acquire and assemble the light probe image.
2. Model the geometry and reflectance of the scene.
3. Map the light probe to an emissive surface surrounding the scene.
4. Render the scene as illuminated by the IBL environment.
5. Postprocess and tone map the renderings.

11.1.1 ACQUIRE AND ASSEMBLE THE LIGHT PROBE

The lighting environment for RNL was acquired in the late afternoon using a 3-inch chrome bearing and a digital camera. The mirrored ball and a digital video camera were placed on tripods about 4 feet from each other and 3.5 feet off the ground.[1] The digital video camera was zoomed until the sphere filled the frame, and the focus was set so that the reflected image was sharp. The aperture of the camera was narrowed to f/8 to allow for sufficient depth of field, and an HDR image series was acquired with shutter speeds varying from 1/4 to 1/10,000 s, spaced one stop apart.

1 Many tripods allow the center pole (the vertical pole to which the tripod head is attached) to be removed from the legs and reinserted upside-down, leaving the end of the pole pointing up and able to accommodate a mirrored sphere.

To cover the scene with better sampling, a second series was acquired after having moved the camera 90° around to see the ball from the side (see Section 11.2.1). The process took only a few minutes, and the resulting image series can be seen in Figures 11.2(a) and (c).

Mirrored spheres reflect nearly the entire environment they are in — not, as sometimes assumed, just the hemisphere looking back in the direction of the camera. This follows from the basic mirror formula that the angle of incidence is equal

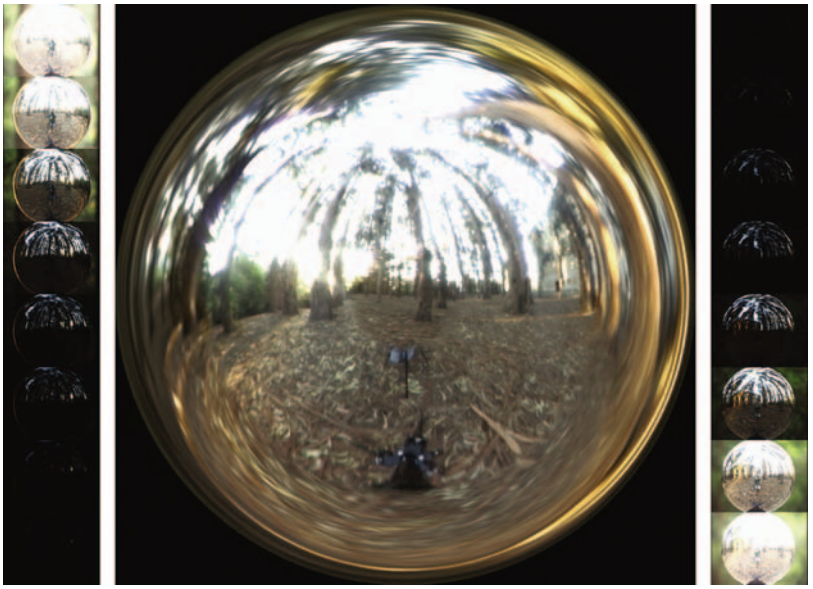


FIGURE 11.2 *The two HDRI series (a and c) used to capture the illumination in the eucalyptus grove for RNL, and the resulting combined light probe image (b), converted into the angular map format.*

to the angle of reflection: rays near the outer edge of a sphere's image have an angle of reflection toward the camera that nears 90°, and thus their angle of incidence also nears 90°. Thus, the ray's angle of incidence relative to the camera nears 180°, meaning that the rays originate from nearly the opposite side of the sphere relative to the camera.

Each image series of the sphere was converted into an HDR image using the HDR image assembly algorithm in Debevec and Malik [62] (see Chapter 4), and the images were saved in RADIANCE's native HDR image format (see Chapter 3). The algorithm derived the response curve of the video camera and produced HDR images where the pixel values were proportional to the light values reflected by the mirrored sphere. The total dynamic range of the scene was approximately 5000:1, measuring from the dark shadows beneath the bushes to the bright blue of the sky and the thin white clouds lit from behind by the sun. As another measure of the range, the brightest pixel values in the sky and cloud regions were some 150 times the average level of light in the scene. The two views of the sphere were combined (using techniques presented in Section 11.2.1) and mapped into the angular map space (described in Section 11.3.2) to become the light probe image seen in Figure 11.2(b).

11.1.2 MODEL THE GEOMETRY AND REFLECTANCE OF THE SCENE

The RNL scene's spheres, stands, and pedestal were modeled using RADIANCE's standard scene primitives and generators. Each sphere was given a different material property with different colors of glass, metal, and plastic. The pedestal itself was texture mapped with a polished marble texture. The scene specification files are included on the companion DVD-ROM as `rnl_scene.rad` and `gensup.sh`.

11.1.3 MAP THE LIGHT PROBE TO AN EMISSIVE SURFACE SURROUNDING THE SCENE

In IBL, the scene is surrounded (either conceptually or literally) by a surface onto which the light probe image is mapped. In the simplest case, this surface is an infinite sphere. The RNL animation used a large but finite inward-pointing cube, positioned so that the bottom of the pedestal sat centered on the bottom of the cube

(Figure 11.3). The light probe image was mapped onto the inner surfaces of the cube so that from the perspective of the top of the pedestal, the light from the environment would come from substantially the same directions as it would have in the forest. The RADIANCE shading language was sufficiently general to allow this mapping to be specified in a straightforward manner. When a ray hits a surface, it reports the three-dimensional (3D) point of intersection $P = (P_x, P_y, P_z)$ to user-supplied equations that compute the texture map coordinate for that point of the surface. In the RNL scene, the top of the pedestal was at the origin, and thus the direction vector into the probe was simply the vector pointing toward P. From this direction, the (u, v) coordinates for the corresponding pixel in the light probe image are computed using the angular map equations in Section 11.3.2. These calculations are specified in the file `angmap.cal` included on the companion DVD-ROM.

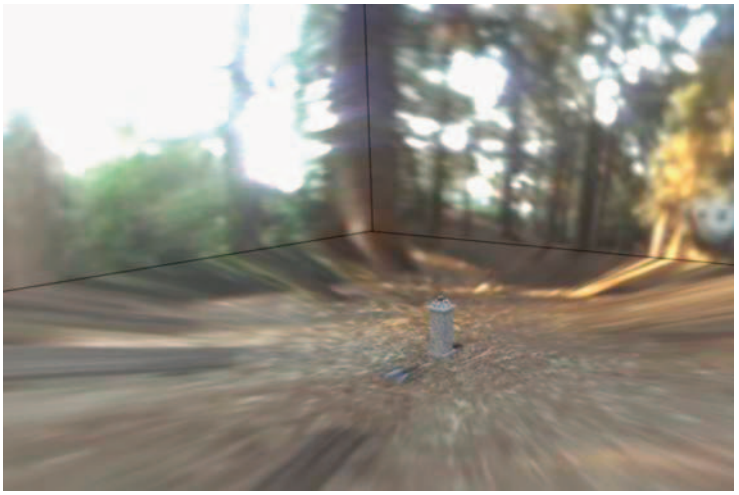


FIGURE 11.3 *The RNL pedestal is seen within the large surrounding box, texture mapped with the eucalyptus grove light probe image.*

In IBL, the surface surrounding the scene is specified to be *emissive*, so that its texture is treated as a map of light emanating from the surface rather than a map of surface reflectance. In RADIANCE, this is done by assigning this environment the *glow* material, which tells the renderer that once a ray hits this surface, the radiance along the ray should be taken directly as the HDR color in the image, rather than the product of the texture color and the illumination incident on it. When the environment surface is viewed directly (as in most of Figure 11.3), it appears as an image-based rendering with the same pixel colors as in the original light probe image.

11.1.4 RENDER THE SCENE AS ILLUMINATED BY THE IBL ENVIRONMENT

With the scene modeled and the light probe image mapped onto the surrounding surface, RADIANCE was ready to create the renderings using IBL. Appropriate rendering parameters were chosen for the number of rays to be used per pixel, and a camera path was animated to move around and within the scene. RADIANCE simulated how the objects would look as if illuminated by the light from the environment surrounding them. Some renderings from the resulting image sequence are shown in Figure 11.4. This lighting process is explained in further detail in Section 11.4.

11.1.5 POSTPROCESS THE RENDERINGS

A final step in creating an IBL rendering or animation is to choose how to tone map the images for display. RADIANCE and many recent rendering systems can output their renderings as HDR image files. Because of this, the RNL renderings exhibited the full dynamic range of the original lighting environment, including the very bright areas seen in the sky and in the specular reflections of the glossy spheres. Because the renderings exhibit a greater dynamic range than can be shown on typical displays, some form of tone mapping is needed to produce the final displayable images. The most straightforward method of tone mapping is to pick a visually pleasing exposure factor for the image, truncate the bright regions to the

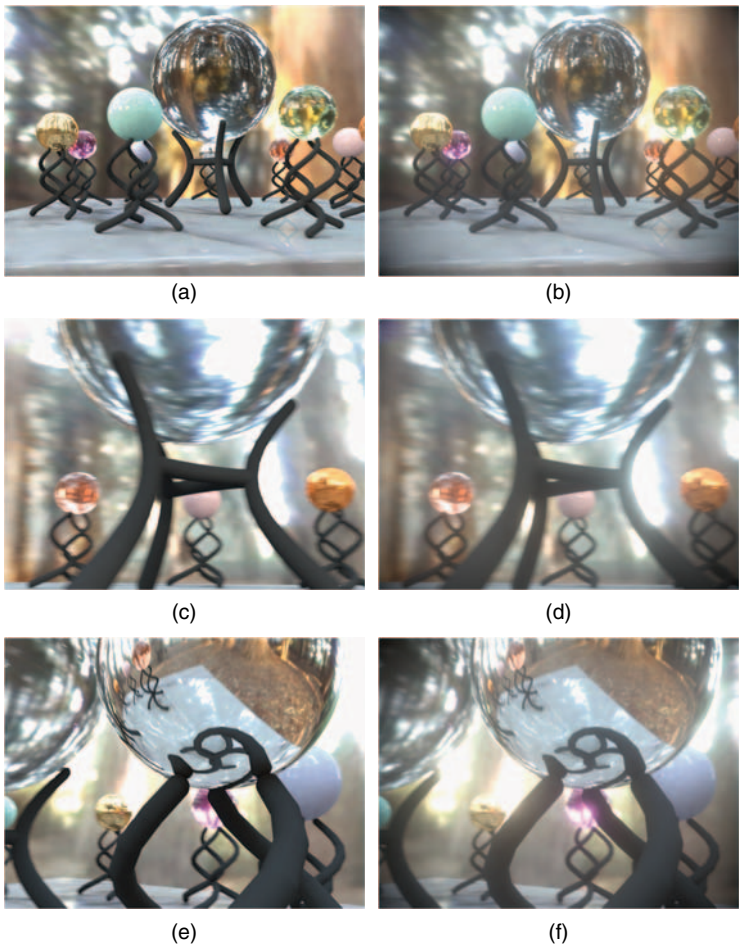


FIGURE 11.4 *Three frames from RNL. Rendered frames (a, c, and e) before postprocessing. After postprocessing (b, d, and f) as described in Section 11.1.5.*

maximum "white" value of the display, and apply any needed compensation for the response curve of the display (most commonly, applying a gamma-correction function). With this technique, values below the white point are reproduced accurately, but everything above the white point is clipped.

Chapters 6 through 8 discussed several tone-reproduction operators that reduce the dynamic range of an image in a natural way to fall within the range of the display, any of which could be used for postprocessing IBL images. Another approach to postprocessing HDR images is to simulate some of the optical imperfections of a real camera system that communicate the full dynamic range of bright regions through blooming and vignetting effects. Such operators often work well in conjunction with IBL rendering because, like IBL, they are designed to simulate the appearance of photographic imagery.

Let's first examine how a blurring operator can communicate the full dynamic range of a scene even on an LDR display. The top of Figure 11.5 shows two bright square regions in an image. In the HDR image file, the right-hand square is six times brighter than the left (as seen in the graph below the squares). However, because the maximum "white" point of the display is below the brightness of the dimmer square, the displayed squares appear to be the same intensity. If we apply a Gaussian blur convolution to the HDR pixel values, the blurred squares appear very different, even when clipped to the white point of the display. The dim blurred square now falls considerably below the white point, whereas the middle of the bright blurred square still exceeds the range of the display. The brighter region also appears larger, even though the regions were originally the same size and are filtered with the same amount of blur. This effect is called "blooming."

Similar blooming effects are seen frequently in real photographs, in which blur can be caused by any number of factors, including camera motion, subject motion, image defocus, "soft focus" filters placed in front of the lens, dust and coatings on lens surfaces, and scattering of light within the air and image sensor. Figure 11.6(a) shows an image acquired in HDR taken inside Stanford's Memorial church. When a clipped LDR version of the image is blurred horizontally (Figure 11.6[b]), the bright stained-glass windows become noticeably darker. When the HDR version of the image is blurred with the same filter (Figure 11.6[c]), the windows appear as vibrant bright streaks, even when clipped to the white point of the display. In addition to the HDR image series, the photographer also acquired a motion-blurred

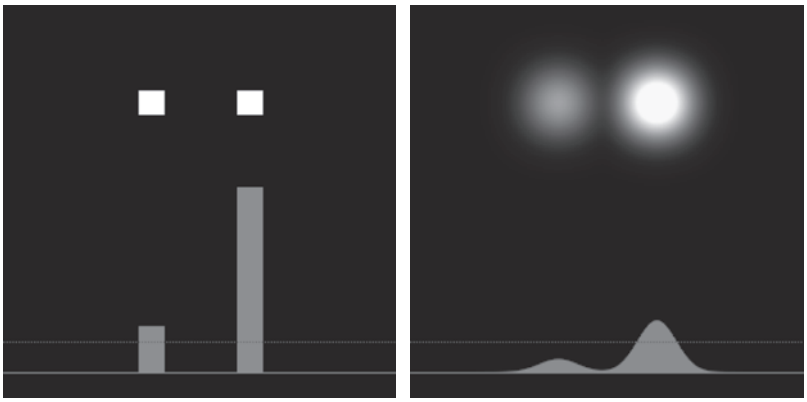


FIGURE 11.5 *Two bright image squares (a) with pixel values 1.5 and 6 are shown on a display with a white point of 1. Because of clipping, the squares appear the same. A graph of pixel intensity for a scan line passing through the centers of the squares is shown below, with "white" indicated by the dotted line. After applying a Gaussian blur (b), the blurred squares are very different in appearance, even though they are still clipped to the display white point.*

version of the church interior by rotating the camera on the tripod during a half-second exposure (Figure 11.6[d]). The bright streaks in this real blurred image (although not perfectly horizontal) are very similar to the streaks computed by the HDR blurring process seen in Figure 11.6(c), and dissimilar to the LDR blur seen in Figure 11.6(d).

The renderings for RNL were postprocessed using a summation of differently blurred versions of the renderings produced by RADIANCE. Each final image used in the film was a weighted average of several differently blurred versions of the image. All of the blur functions used in RNL were Gaussian filters, and their particular mixture is illustrated in Figure 11.7.

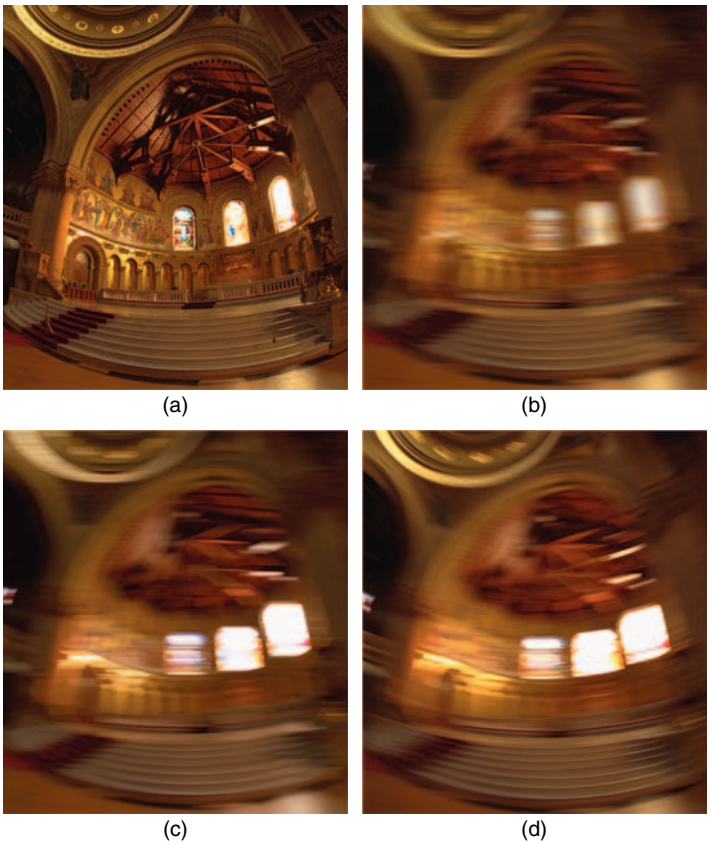


FIGURE 11.6 *An HDR scene inside a church with bright stained-glass windows. (b) Horizontally blurring the clipped LDR image gives the effect of image motion, but it noticeably darkens the appearance of the stained-glass windows. (c) Blurring an HDR image of the scene produces bright and well-defined streaks from the windows. (d) Real motion blur obtained by rotating camera during the exposure validates the HDR blurring simulated in image (**c**).*

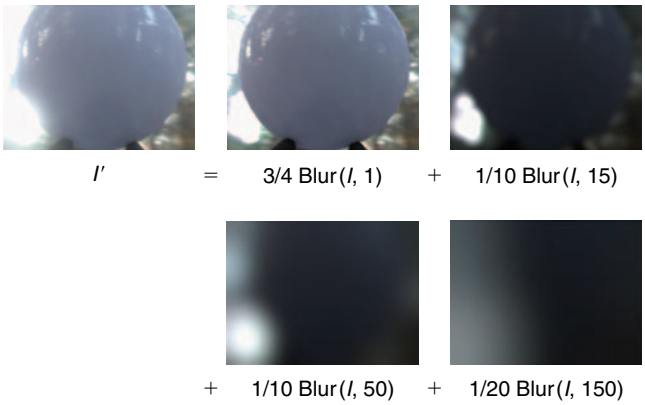


FIGURE 11.7 *The mix of blur filters used to postprocess the RNL frames. Blur*(I, σ) *indicates a Gaussian blur with a standard deviation of* σ *pixels.*

With the right techniques, such blurring processes can be performed very efficiently, even though convolving images with wide filters is normally computationally expensive. First, Gaussian filters are *separable*, so that the filter can be performed as a one-dimensional (1D) Gaussian blur in x followed by a 1D Gaussian blur in y. Second, a Gaussian blur can be closely approximated as several successive box blurs. Finally, the wider blurs can be closely approximated by applying correspondingly narrower blurring filters to a low-resolution version of the image, and then upsampling the small blurred version. With these enhancements, efficient implementations of such techniques can be achieved on modern GPUs (Graphics Processing Unit) [223], and considerably more elaborate lens flare effects can be performed in real time [149].

Postprocessed frames from RNL can be seen in the right-hand column of Figure 11.4. Because the final postprocessed images are 75% composed of the original renderings with just a slight blur applied, the original image detail is still

evident. However, because of the other blurred versions added, the bright parts of the environment and their specular reflections tend to bloom in the final renderings. Often, as in Figure 11.7, the bloom from bright spots in the environment appears to "wrap" around objects in the foreground. This is a surprisingly natural effect that helps the rendered objects appear to belong within the rest of the scene.

As mentioned previously, this postprocessing of HDR imagery is a form of tone mapping, the effects of which are similar to the results produced by using "soft focus," "mist," and "fog" filters on real camera lenses. The effects are also similar to the effects of the optical imperfections of the human eye. A detailed model of the particular glare and bloom effects produced in the human eye is constructed and simulated by Spencer et al. [303]. In addition, a basic model of human eye glare was used in conjunction with a tone-reproduction operator by Larson et al. [176].

A final subtle effect applied to the renderings in RNL is *vignetting*. Vignetting is the process of gently darkening the pixel values of an image toward its corners, which occurs naturally in many camera lenses (particularly at wide apertures) and is sometimes intentionally exaggerated with an additional mask or iris for photographic effect. Applying this effect to an HDR image before tone mapping the pixel values can also help communicate a greater sense of the range of light in a scene, particularly in animations. With this effect, as a bright region moves from the center of the field of view to the edge, the pixels around it dim. However, its particularly bright pixels will still reach the white point of the display. Thus, different exposures of the scene are revealed in a natural manner simply through camera motion. The effect is easily achieved by multiplying an image by a brightness falloff image such as in Figure 11.8(a). Figures 11.8(b) and (c) show a rendering from RNL before and after vignetting.

As an early IBL example, RNL differed from most CG animations in that designing the lighting in the scene was a matter of choosing real light from a real location rather than constructing the light as an arrangement of computer-generated light sources. Using global illumination to simulate the IBL naturally produced shading, highlights, refractions, and shadows that were consistent with one another and with the environment surrounding the scene. With traditional CG lighting, such an appearance would have been difficult to achieve.

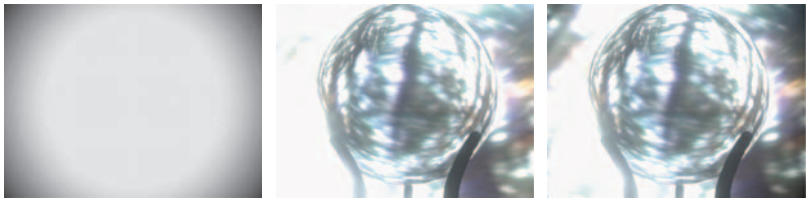


FIGURE 11.8 Through the brightness falloff function shown in image (a), a frame from the RNL animation is seen (b) before and (c) after vignetting. In this image, this operation darkens the right-side image corners, reveals detail in the slightly overexposed upper-left corner, and has little effect on the extremely bright lower-left corner. In motion, this vignetting effect helps communicate the HDR environment on an LDR display.

11.2 CAPTURING LIGHT PROBE IMAGES

Moving on from our basic example, the next few sections present the key stages of IBL with greater generality and detail, beginning with the process of lighting capture. Capturing the incident illumination at a point in space requires taking an image with two properties. First, it must see in all directions, in that light coming from anywhere can affect the appearance of an object. Second, it must capture the full dynamic range of the light within the scene, from the brightest concentrated light sources to the dimmer but larger areas of indirect illumination from other surfaces in the scene. In many cases, the standard HDR photography techniques presented in Chapter 4 satisfy this second requirement. Thus, the remaining challenge is to acquire images that see in all directions, a process known as "panoramic" (or "omnidirectional") photography. There are several methods of recording images that see in all directions, each with advantages and disadvantages. In this section, we describe some of the most common techniques, which include using mirrored spheres, tiled

photographs, fish-eye lenses, and scanning panoramic cameras. This section concludes with a discussion of how to capture light probe images that include a direct view of the sun, which is usually too bright to record with standard HDR image capture techniques.

11.2.1 PHOTOGRAPHING A MIRRORED SPHERE

The technique used to capture the RNL light probe was to photograph a mirrored sphere placed in the scene where it is desired to capture the illumination. Using mirrored spheres to obtain omnidirectional reflections of an environment was first used for *environment mapping* [222,361,113] (described in Section 11.7.1), where such images are directly texture mapped onto surfaces of objects. The main benefit of photographing a mirrored sphere is that it reflects very nearly the entire environment in a single view. Aside from needing two tripods (one for the sphere and one for the camera), capturing a light probe image with this technique can be fast and convenient. Mirrored spheres are inexpensively available as 2- to 4-inch diameter chrome ball bearings (available from the McMaster–Carr catalog, http://www.mcmaster.com), 6- to 12-inch mirrored glass lawn ornaments (available from Baker's Lawn Ornaments, http://www.bakerslawnorn.com), and Chinese meditation balls (1.5–3 inches). Dubé juggling equipment (http://www.dube.com) sells polished hollow chrome spheres from $2\frac{1}{6}$ to $2\frac{7}{8}$ inches in diameter. Professionally manufactured mirrored surfaces with better optical properties are discussed at the end of this section.

There are several factors that should be considered when acquiring light probe images with a mirrored sphere. These are discussed in the sections that follow.

Framing and Focus First, it is desirable to have the sphere be relatively far from the camera to minimize the size of the camera's reflection and to keep the view nearly orthographic. To have the sphere be relatively large in the frame, it is necessary to use a long-focal-length lens. Many long lenses have difficulty focusing closely on small objects, and thus it may be necessary to use a $+1$ diopter close-up filter (available from a professional photography store) on the lens to bring the sphere into focus. The image of the sphere usually has a shallow depth of field, especially when a close-up filter is used, and thus, it is often necessary to use an aperture of f/8 or smaller to bring the full image into focus.

Blind Spots There are several regions in a scene that are usually not captured well by a mirrored sphere. One is the region in front of the sphere, which reflects the camera and often the photographer. Another is the region directly behind the sphere, which is reflected by a thin area around the edge of the sphere. The last is a strip of area from straight down and connecting to the area straight back, which usually reflects whatever supports the sphere. For lighting capture, these effects are easily minimized by orienting the camera so that no photometrically interesting areas of the scene (e.g., bright light sources) fall within these regions. However, it is sometimes desirable to obtain clear images of all directions in the environment, for example, when the light probe image itself will be seen in the background of the scene. To do this, one can take two HDR images of the mirrored sphere, with the second rotated 90° around from the first. In this way, the poorly represented forward and backward directions of one sphere correspond to the well-imaged left and right directions of the other, and vice versa. The two images taken for the RNL light probe are shown in Figure 11.9. Each image slightly crops the top and bottom of the sphere, which was done intentionally to leverage the fact that these areas belong to the rear half of the environment that appears in the other sphere image. Using an HDR image-editing program such as HDR Shop [317], these two images can be combined into a single view of the entire environment that represents all directions well except straight down. If needed, this final area could be filled in from a photograph of the ground or through manual image editing.

Calibrating Sphere Reflectivity It is important to account for the fact that mirrored spheres are generally not optically perfect reflectors. Although the effect is often unnoticed, ball bearings typically reflect only a bit more than half of the light hitting them. In some IBL applications, the lighting is captured using a mirrored sphere, and the background image of the scene is photographed directly. To correct the sphere image so that it is photometrically consistent with the background, we need to measure the reflectivity of the sphere. This can be done using a setup such as that shown in Figure 11.10. In this single photograph, taken with a radiometrically calibrated camera, the indicated patch of diffuse paper is reflected in the ball. We can thus divide the average pixel value of the patch in the reflection by the average pixel value in the direct view to obtain the sphere's percent reflectivity in each of the red, green, and blue channels. A typical result would be $(0.632, 0.647, 0.653)$. Often, these three numbers will be slightly different, indicating that the sphere tints

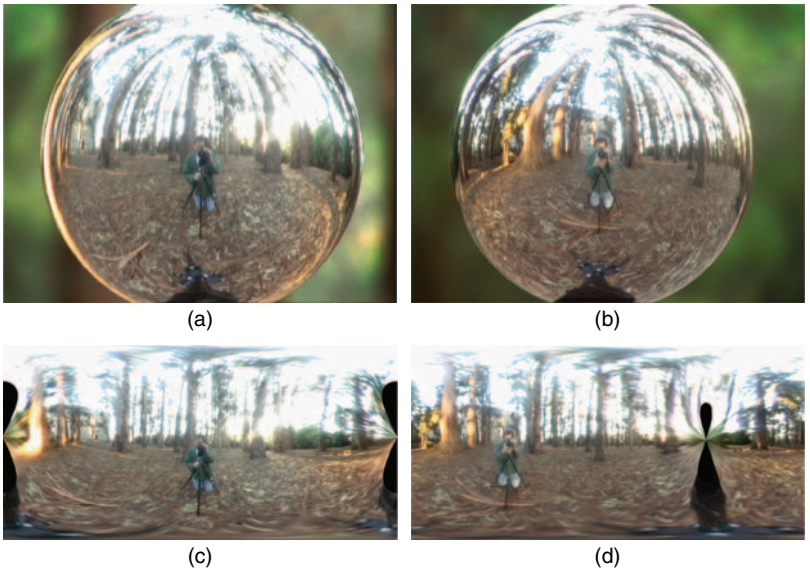


FIGURE 11.9 *Acquiring a light probe with a mirrored sphere. (a) and (b) show two images of the sphere, taken 90° apart. (c) and (d) show the mirrored sphere images transformed into latitude–longitude mappings. In (d), the mapping has been rotated 90° to the left to line up with the mapping of (c). The black teardrop shapes correspond to the cropped regions at the top and bottom of each sphere, and the pinched area between each pair of drops corresponds to the poorly sampled region near the outer edge of the sphere image. Each sphere yields good image data where the other one has artifacts, and combining the best regions of each can produce a relatively seamless light probe image, as in Figure 11.2.*

the incident light. The light probe image can be corrected to match the background by dividing its channels by each of these numbers.

Nonspecular Reflectance Mirrored spheres usually exhibit a faint diffuse or rough specular component because of microscopic scratches and deposits on their surface.

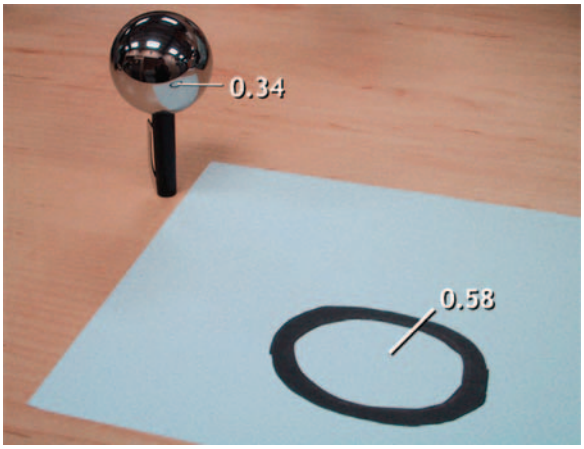


FIGURE 11.10 *The reflectivity of a mirrored sphere can be determined by placing the sphere near an easy-to-identify part of a diffuse surface, and comparing its brightness seen directly to its appearance as reflected in the sphere. In this case, the sphere is 59% reflective.*

It is best to keep the spheres in cloth bags to minimize such scratching, as well as to keep them dry so that the surface does not oxidize. A slightly rough specular component usually makes little difference in how light probe images illuminate CG objects, but when viewed directly, the reflected image may lack contrast in dark regions and exhibit bright flares around the light sources. If an application requires a near-perfectly shiny mirrored surface, one can have a glass or metal sphere or hemisphere specially coated with a thin layer of aluminum by an optical coating company (only half a sphere can be photographed at once, and thus in practice, a hemisphere can also be used as a light probe). Such coated optics yield extremely clear specular reflections and can be up to 91% reflective. Some experiments in capturing the sky using such optics can be found in Stumpfel [309].

Polarized Reflectance Mirrored spheres behave somewhat unexpectedly with respect to polarization. Light that reflects from a sphere at angles next to the

outer rim becomes polarized, an effect characterized by Fresnel's equations [98]. Camera sensors generally record light irrespective of polarization, so this itself is not a problem.[2] However, for the same reasons, polarized light reflecting from a mirrored sphere can appear either too bright or too dim compared with being viewed directly. This is a significant effect in outdoor environments, where the scattered blue light of the sky is significantly polarized. This problem can be substantially avoided by using highly reflective mirror coatings (as discussed previously).

Image Resolution It can be difficult to obtain a particularly high-resolution image of an environment as reflected in a mirrored sphere, because only one image is used to cover a fully spherical field of view. For lighting CG objects, the need for highly detailed light probe images is not great; only large and very shiny objects reflect light in a way in which fine detail in an environment can be noticed. However, for forming virtual backgrounds behind CG objects, it is often desirable to have higher-resolution imagery. In the RNL animation, the low resolution of the light probe image used for the background produced the reasonably natural appearance of a shallow depth of field, even though no depth of field effects were simulated.

Photographing a mirrored sphere is an example of a *catadioptric* imaging system in that it involves both a lens and a reflective surface. In addition to mirrored spheres, other shapes can be used that yield different characteristics of resolution, depth of field, and field of view. One example of a well-engineered omnidirectional video camera covering slightly over a hemispherical field of view is presented in Nayer [236]. Nonetheless, for capturing illumination wherein capturing the full sphere is more important than high image resolution, mirrored spheres are often the most easily available and convenient method.

11.2.2 TILED PHOTOGRAPHS

Omnidirectional images can also be captured by taking a multitude of photographs looking in different directions and "stitching" them together—a process made familiar by QuickTime VR panoramas [36]. This technique can be used to assemble

2 Although sensors tend to detect different polarization directions equally, wide-angle lenses can respond differently according to polarization for regions away from the center of the image.

remarkably high-resolution omnidirectional images using a standard camera and lens. Unfortunately, the most commonly acquired panoramas see all the way around the horizon but only with a limited vertical field of view. For capturing lighting, it is important to capture imagery looking in *all* directions, particularly upward, because this is often where much of the light comes from. Images taken to form an omnidirectional image will align much better if the camera is mounted on a nodal rotation bracket, which can eliminate viewpoint parallax between the various views of the scene. Such brackets are available commercially from companies such as Kaidan (http://www.kaidan.com). Some models allow the camera to rotate around its nodal center for both the horizontal and vertical axes.

Figure 11.11 shows tone-mapped versions of the source HDR images for the "Uffizi Gallery" light probe image [57], which was acquired as an HDR tiled panorama. These images were aligned by marking pairs of corresponding points between the original images and then solving for the best 3D rotation of each image to minimize the distance between the marked points. The images were then blended across their edges to produce the final full-view latitude–longitude mapping. This image was used as the virtual set and lighting environment for the middle sequence of the animation *Fiat Lux*.

A more automatic algorithm for assembling such full-view panoramas is described in [315], and commercial image-stitching products such as QuickTime VR (http://www.apple.com/quicktime/qtvr/) and Realviz Stitcher (http://www.realviz.com) allow one to interactively align images taken in different directions to produce full-view panoramas in various image formats. Unfortunately, at the time of writing, no commercial products natively support stitching HDR images. Digital photographer Greg Downing (http://www.gregdowning.com) has described a process [70] for stitching each set of equivalent exposures across the set of HDR images into its own panorama, and then assembling this series of LDR panoramas into a complete light probe image. The key is to apply the same alignment to every one of the exposure sets: If each set were aligned separately, there would be little chance of the final stitched panoramas aligning well enough to be assembled into an HDR image. To solve this problem, Downing uses Realviz Stitcher to align the exposure level set with the most image detail and saves the alignment parameters in a way in which they can be applied identically to each exposure level across the set of views. These differently exposed LDR panoramas can then be properly assembled into an HDR panorama.

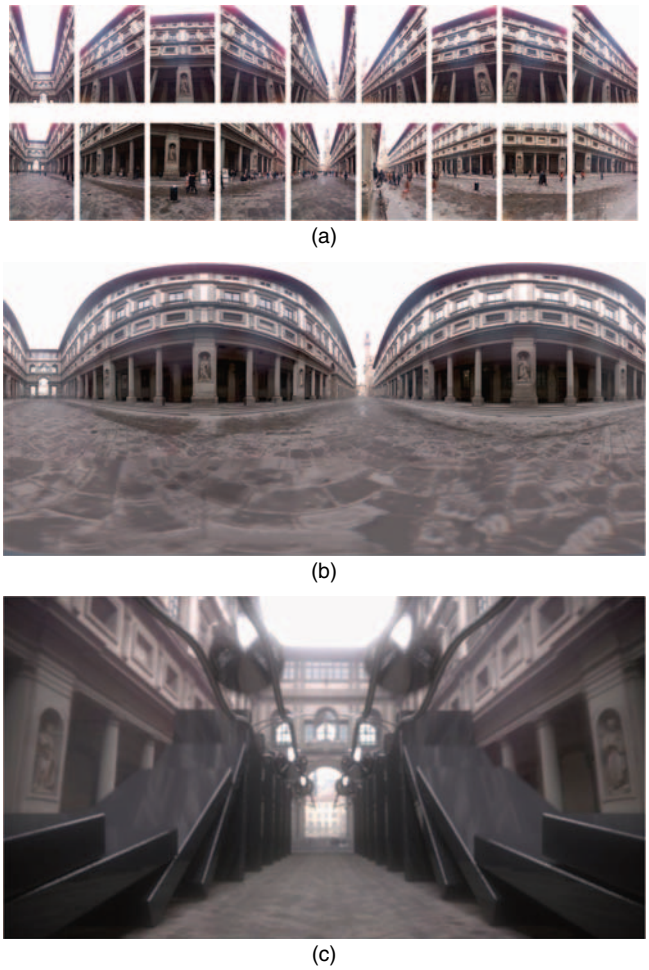


FIGURE 11.11 (a) The Uffizi light probe was created from an HDR panorama with two rows of nine HDR images. (b) The assembled light probe in latitude–longitude format. (c) Synthetic objects added to the scene, using the light probe as both the virtual background and the lighting environment.

11.2.3 FISH-EYE LENSES

Fish-eye lenses are available for most single-lens reflex cameras and are capable of capturing 180° single view. As a result, they can cover the full view of an environment in as few as two images. In Greene [113], a fish-eye photograph of the sky was used to create the upper half of a cube map used as an environment map. Although fish-eye lens images are typically not as sharp as regular photographs, light probe images obtained using fish-eye lenses are usually of higher resolution than those obtained by photographing mirrored spheres. A challenge in using fish-eye lenses is that not all 35-mm digital cameras capture the full field of view of a 35-mm film camera as a result of having a smaller image sensor. In this case, the top and bottom of the circular fish-eye image are usually cropped off. This can require taking additional views of the scene to cover the full environment. Fortunately, recently available digital cameras, such as the Canon EOS 1Ds and the Kodak DCS 14n, have image sensor chips that are the same size as 35-mm film (and no such cropping occurs).

Fish-eye lenses can exhibit particularly significant radial intensity falloff, also known as *vignetting*. As with other lenses, the amount of falloff tends to increase with the size of the aperture being used. For the Sigma 8-mm fish-eye lens, the amount of falloff from the center to the corners of the image is more than 50% at its widest aperture of f/4. The falloff curves can be calibrated by taking a series of photographs of a constant-intensity source at different camera rotations and fitting a function to the observed image brightness data. This surface can be used to render an image of the *flat-field response* of the camera. With this image, any HDR image obtained with the camera can be made radiometrically consistent across its pixels by dividing by the image of the flat-field response. An example of this process is described in more detail in [310].

11.2.4 SCANNING PANORAMIC CAMERAS

Scanning panoramic cameras are capable of capturing particularly high-resolution omnidirectional HDR images. These cameras use narrow image sensors that are typically three pixels wide and several thousand pixels tall. The three columns of pixels are filtered by red, green, and blue filters, allowing the camera to sense color. A precise motor rotates the camera by 360° over the course of a few seconds to a few

(a)

(b)

FIGURE 11.12 *(a) The Spheron scanning panoramic camera. (b) A tone-mapped version of a high-resolution omnidirectional HDR image taken with the camera. (Panorama courtesy of Ted Chavalas of Panoscan, Inc.)*

minutes, capturing a vertical column of the panoramic image many times per second. When a fish-eye lens is used, the full 180° vertical field of view can be captured from straight up to straight down. Two cameras based on this process are made by Panoscan and Spheron VR (Figure 11.12).

Trilinear image sensors, having far fewer pixels than area sensors, can be designed with more attention given to capturing a wide dynamic range in each exposure. Nonetheless, for IBL applications in which it is important to capture the full range of light, including direct views of concentrated light sources, taking multiple exposures is usually still necessary. The Panoscan camera's motor is able to precisely rewind and repeat its rotation, allowing multiple exposures to be taken and assembled into an HDR image without difficulty. The Spheron VR camera (see also Section 5.11.5) can be ordered with a special HDR feature in which the image sensor rapidly captures an HDR series of exposures for each column of pixels as the camera head rotates. These differently exposed readings of each pixel column can be assembled into HDR images from just one rotation of the camera.

For these cameras, the speed of scanning is limited by the amount of light in the scene. Suppose that at the chosen f-stop and ISO[3] setting it takes 1/125 s to obtain a proper exposure of the shadows and midtones of a scene. If the image being acquired is 12,000 pixels wide, the camera must take at least 96 s to scan the full panorama. Fortunately, shooting the rest of the HDR image series to capture highlights and light sources takes considerably less time because each shutter speed is shorter, typically at most one-fourth the length for each additional exposure. Although capturing lighting with scanning panoramic cameras is not instantaneous, the resulting lighting environments can have extremely detailed resolution. The high resolution also enables using these images as background plates or to create image-based models for 3D virtual sets.

11.2.5 CAPTURING ENVIRONMENTS WITH VERY BRIGHT SOURCES

For IBL, it is important that the acquired light probe images cover the full dynamic range of light in the scene up to and including light sources. If 99.99% of a light probe image is recorded properly but 0.01% of the pixel values are saturated, the light captured could still be very inaccurate depending on how bright the saturated pixels really should have been. Concentrated light sources are often significantly brighter than the average colors within a scene. In a room lit by a bare light bulb, the light seen reflecting from tens of square meters of ceiling, floor, and walls originates

3 The International Organization for Standardization has specified a standard scale for film speed, called ISO 5800:1987 which in photography is commonly referred to as "ISO".

from just a few square millimeters of light bulb filament. Because of such ratios, light sources are often thousands, and occasionally millions, of times brighter than the rest of the scene.

In many cases, the full dynamic range of scenes with directly visible light sources can still be recovered using standard HDR photography techniques, in that camera shutter speeds can usually be varied down to 1/4000 s or shorter, and small apertures can be used as well. Furthermore, modern lighting design usually avoids having extremely concentrated lights (such as bare filaments) in a scene, preferring to use globes and diffusers to more comfortably spread the illumination over a wider area. However, for outdoor scenes, the sun is a light source that is both very bright and very concentrated. When the sun is out, its brightness can rarely be recorded using a typical camera even using the shortest shutter speed, the smallest aperture, and the lowest sensor gain settings. The sun's brightness often exceeds that of the sky and clouds by a factor of 50,000, which is difficult to cover using varying shutter speeds alone.

Stumpfel et al. [310] presented a technique for capturing light from the sky up to and including the sun. To image the sky, the authors used a Canon EOS 1Ds digital camera with a Sigma 8-mm fish-eye lens facing upward on the roof of an office building (Figure 11.13[a]). The lens glare caused by the sun was minor, which was verified by photographing a clear sky twice and blocking the sun in one of the images. For nearly the entire field of view, the pixel values in the sky were within a few percentage points of each other in both images.

As expected, the sun was far too bright to record even using the camera's shortest shutter speed of 1/8000 s at f/16, a relatively small aperture.[4] The authors thus placed a Kodak Wratten 3.0 neutral density (ND) filter on the back of the lens to uniformly reduce the light incident on the sensor by a factor of 1000.[5] ND filters are often not perfectly neutral, giving images taken through them a significant color cast. The authors calibrated the transmission of the filter by taking HDR images of a scene with and without the filter, and divided the two images to determine the filter's transmission in the red, green, and blue color channels.

4 The authors observed unacceptably pronounced star patterns around the sun at the smaller apertures of f/22 and f/32 as a result of diffraction of light from the blades of the iris.

5 Because the fish-eye lens has such a wide-angle view, filters are placed on the back of the lens using a small mounting bracket rather than on the front.

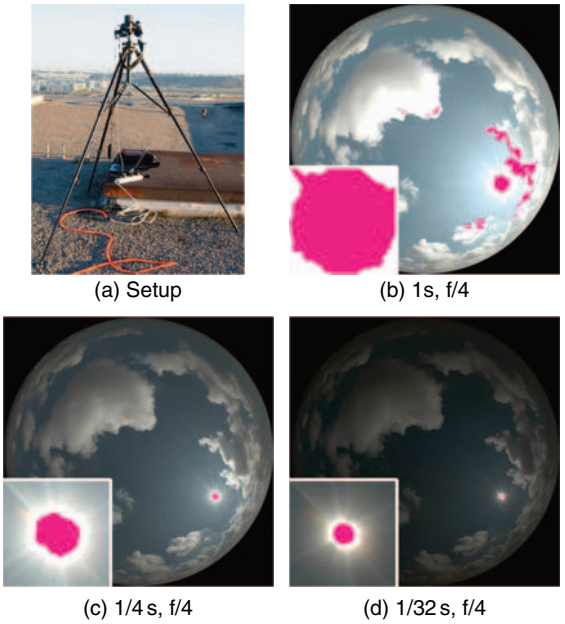
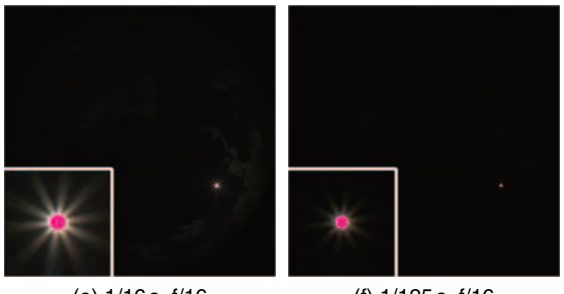


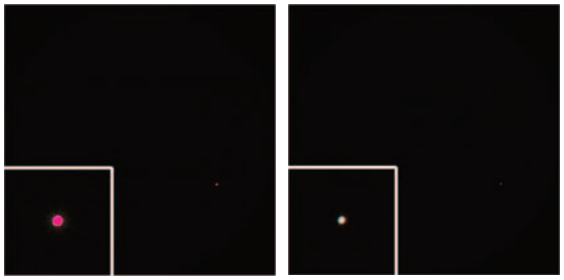
FIGURE 11.13 (a) *A computer-controlled camera with a fish-eye lens placed on a roof to capture HDR images of the sky. A color-corrected HDR image series (b through h) of a sunny sky taken using varying shutter speed, aperture, and a 3.0 ND filter. The inset of each frame shows a small area around the sun, with saturated pixels shown in pink. Only the least exposed image (h) accurately records the sun's intensity without sensor saturation.*

All images subsequently taken through the filter were scaled by the inverse of these transmission ratios.

Having the 3.0 ND filter on the lens made it possible to image the sun at 1/8000 s at f/16 without saturating the sensor (see Figure 11.13[a]), but it made the sky and clouds require an undesirably long exposure time of 15 s. To solve this problem, the



(e) 1/16 s, f/16 (f) 1/125 s, f/16



(g) 1/1000 s, f/16 (h) 1/8000 s, f/16

FIGURE 11.13 (*continued*)

authors used a laptop computer to control the camera so that both the shutter speed and the aperture could be varied during each HDR image sequence. Thus, the series began at f/4 with exposures of 1, 1/4, and 1/32 s and then switched to f/16 with exposures of 1/16, 1/125, 1/1000, and 1/8000 s. Images from such a sequence are seen in Figure 11.13(b) through 11.13(h). For presunrise and postsunrise images, the f/16 images were omitted and an additional exposure of 4 s at f/4 was added to capture the dimmer sky of dawn and dusk.

Creating HDR images using images taken with different apertures is slightly more complicated than usual. Because different apertures yield different amounts of lens

vignetting, each image needs to be corrected for its aperture's flat-field response (see Section 11.2.3) before the HDR assembly takes place. In addition, whereas the actual exposure ratios of different camera shutter speeds typically follow the expected geometric progression (1/30 s is usually precisely half the exposure of 1/15 s; 1/15 s is usually precisely half the exposure of 1/8 s), aperture transmission ratios are less exact. In theory, images taken at f/4 should receive 16 times the exposure of images taken at f/16, but generally do not. To test this, the authors took images of a constant intensity light source at both f/4 and f/16 and compared the pixel value ratios at the center of the image, measuring a factor of 16.63 rather than 16, and compensated for this ratio accordingly.

When the sun was obscured by clouds or was low in the sky, the images with the shortest exposure times were not required to cover the full dynamic range. The authors discovered they could adaptively shoot the HDR image sequence by programming the laptop computer to analyze each image in the series after it was taken. If one of the images in the series was found to have no saturated pixels, no additional photographs were taken.

Indirectly deriving sun intensity from a diffuse sphere rather than using specialized HDR photography to directly capture the sun's intensity, we can estimate the sun's intensity based on the appearance of a diffuse object within the scene. In particular, we can use a mirrored sphere to image the ground, sky, and clouds, and a diffuse sphere to indirectly measure the intensity of the sun. Such an image pair is shown in Figure 11.14.

The mirrored sphere contains accurate pixel values for the entire sky and surrounding environment, except for the region of sensor saturation near the sun. Because of this, the clipped light probe will not accurately illuminate a synthetic object as it would really appear in the real environment; it would be missing the light from the sun. We can quantify the missing light from the sun region by comparing the real and synthetic diffuse sphere images.

To perform this comparison, we need to adjust the images to account for the reflectivity of each sphere. For the diffuse sphere, a gray color can be preferable to white because it is less likely to saturate the image sensor when exposed according to the average light in the environment. The reflectivity of the paint can be measured by painting a flat surface with the same paint and photographing this sample (with a radiometrically calibrated camera) in the same lighting and orientation as a flat surface of known reflectance. Specialized *reflectance standards* satisfying

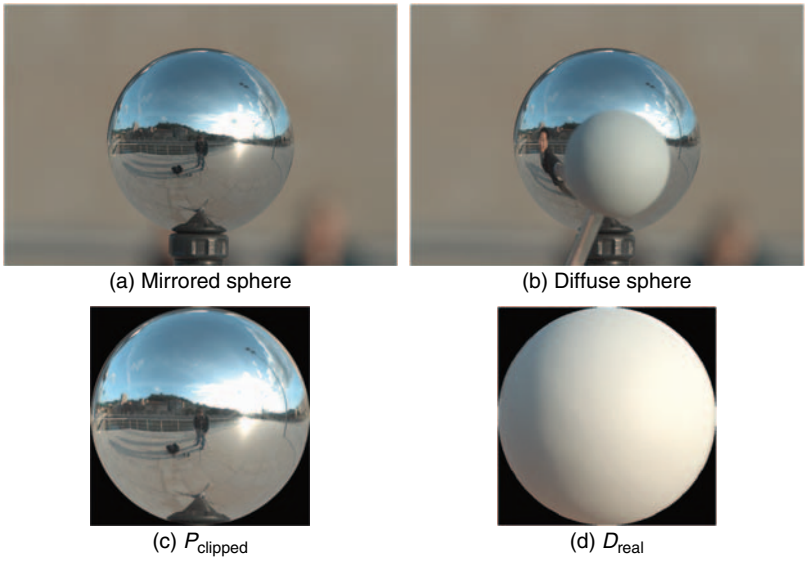


FIGURE 11.14 *(a) A mirrored sphere photographed as a single LDR exposure with direct sunlight. The bright sun becomes a saturated region of pixels clipped to white. (b) A gray diffuse sphere photographed in the same illumination, held several inches in front of the mirrored sphere. (c) The cropped, reflectance-calibrated, and clipped mirrored sphere image* P_{clipped}. *(d) The cropped and reflectance-calibrated image of the diffuse sphere* D_{real}.

this requirement are available from optical supply companies. These reflect nearly 99% of the incident light — almost perfectly white. More economical reflectance standards are the neutral-toned squares of a Gretag–MacBeth ColorChecker chart (http://www.gretagmacbeth.com), whose reflectivities are indicated on the back of the chart. Let us call the reflectivity of our standard ρ_{standard}, the pixel color of the standard in our image L_{standard}, and the pixel color of the paint sample used

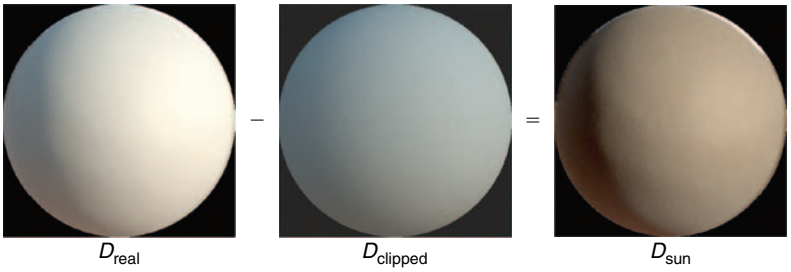


FIGURE 11.15 *The calibrated image of the real diffuse sphere D_{real}, minus the image of a synthetic diffuse sphere lit by the clipped IBL environment D_{clipped}, yields an image of the missing light from the sun D_{sun}.*

for the diffuse sphere L_{paint}. Then, the reflectivity ρ_{paint} of the paint is simply as follows:

$$\rho_{\text{paint}} = \rho_{\text{standard}} \frac{L_{\text{paint}}}{L_{\text{standard}}}.$$

For the diffuse paint used to paint the sphere in Figure 11.14(b), the reflectivity was measured to be (0.320, 0.333, 0.346) in the red, green, and blue channels, indicating that the chosen paint is very slightly bluish. Dividing the pixel values of the diffuse sphere by its reflectivity yields the appearance of the diffuse sphere as if it were 100% reflective, or perfectly white; we call this image D_{real} (see Figure 11.14[d]). Likewise, the image of the mirrored sphere should be divided by the reflectivity of the mirrored sphere as described in Section 11.2.1 and Figure 11.10 to produce the image that would have been obtained from a perfectly reflective mirrored sphere.

With both sphere images calibrated, we can use IBL to light a synthetic white sphere with the clipped lighting environment image, which we call D_{clipped} in Figure 11.15. As expected, this synthetic sphere appears darker than the real sphere in the actual lighting environment D_{real}. If we subtract D_{clipped} from D_{real}, we

obtain an image of the missing reflected light from the sun region (which we call D_{sun}). This operation leverages the additive property of light, described in detail in Section 11.8.

From the previous section, we know that the sun is a 0.53° disk in the sky. To properly add such a source to P_{clipped}, it should be placed in the right direction and assigned the correct radiant intensity. The direction of the sun can usually be estimated from the clipped light probe image as the center of the saturated region in the image. The pixel coordinates (u, v) in this image can be converted to a direction vector (D_x, D_y, D_z) using the ideal sphere-mapping formula discussed in Section 11.3.1. If the saturated region is too large to locate the sun with sufficient precision, it is helpful to have at least one additional photograph of the mirrored sphere taken with a shorter exposure time, or a photograph of a black plastic sphere in which the sun's position can be discerned more accurately. In this example, the sun's direction vector was measured as $(0.748, 0.199, 0.633)$ based on its position in the mirrored sphere image.

To determine the sun's radiance, we know that if the sun alone were to illuminate a white diffuse sphere, the sphere should appear similar to D_{sun}. We begin by creating a sun disk with direction $(0.748, 0.199, 0.633)$, an angular extent of 0.53° diameter and an experimental radiance value of $L_{\text{suntest}} = 46\,700$. The specification for such a light source in a rendering file might look as follows.

```
light sun directional {
  direction 0.748  0.199  0.633
  angle 0.5323
  color 46700 46700 46700
}
```

The value 46,700 is chosen for convenience to be the radiant intensity of a 0.53° diameter infinite light source that illuminates a diffuse white sphere such that its brightest spot (pointing toward the source) has a radiance of 1. Lighting a white sphere with this source produces the image D_{suntest} seen in Figure 11.16.

From the equation shown in Figure 11.16, we can solve for the unknown color α that best scales D_{suntest} to match D_{sun}. This is easily accomplished by dividing the average pixel values of the two images: $\alpha = avg(D_{\text{sun}})/avg(D_{\text{suntest}})$. Then, we can compute the correct radiant intensity of the sun as $L_{\text{sun}} = \alpha L_{\text{suntest}}$.

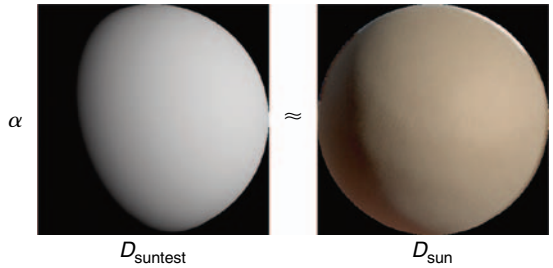


FIGURE 11.16 *Lighting a white sphere (D_{suntest}) from the direction of the sun with an experimental radiant intensity L_{suntest} sets up an equation to determine the missing light from the sun.*

For this example, applying this procedure to each color channel yields $\alpha = (1.166, 0.973, 0.701)$, which produces $L_{\text{sun}} = (54500, 45400, 32700)$. Replacing the L_{suntest} value in the directional light specification file with this new L_{sun} value produces a directional light that models the missing sunlight in the clipped light probe image.

We can validate the accuracy of this procedure by lighting a diffuse sphere with the combined illumination from the clipped probe P_{clipped} and the reconstructed sun. Figures 11.17(a) and 11.17(b) show a comparison between the real diffuse sphere D_{real} and a synthetic diffuse sphere illuminated by the recovered environment. Subtracting (a) from (b) (shown in Figure 11.17[c]) allows us to visually and quantitatively verify the accuracy of the lighting reconstruction. The difference image is nearly black, which indicates a close match. In this case, the root mean squared intensity difference between the real and simulated spheres is less than 2%, indicating a close numeric match as well. The area of greatest error is in the lower left of the difference image, which is a dim beige color instead of black because of unintended bounced light on the real sphere from the person's hand holding the sphere.

FIGURE 11.17 *(a) The calibrated diffuse sphere D_{real} in the lighting environment. (b) A rendered diffuse sphere, illuminated by the incomplete probe $P_{clipped}$ and the recovered sun. (c) The difference between the two. The nearly black image indicates that the lighting environment was recovered accurately.*

At this point, the complete lighting environment is still divided into two pieces: the clipped IBL environment and the directional sun source. To unify them, the sun disk could be rasterized into a mirror reflection image space and then added to the clipped IBL environment. However, as we will see in Section 11.5.1, it can be a great benefit to the rendering process if concentrated lights such as the sun are simulated as direct light sources rather than as part of the IBL environment (because of the sampling problem). Figure 11.38(e) later in this chapter shows a rendering of a collection of CG objects illuminated by the combination of the separate clipped probe and the direct sun source. The appearance of the objects is realistic and consistent with their environment. The rendering makes use of an additional technique (described in Section 11.6) to have the objects cast shadows onto the scene as well.

Images with concentrated light sources such as the sun not only require additional care to acquire but also pose computational challenges for IBL algorithms. Section 11.5 describes why these challenges occur and how they can be solved through importance sampling techniques. Before beginning that topic, we will describe some of the omnidirectional image-mapping formats commonly used to store light probe images.

11.3 OMNIDIRECTIONAL IMAGE MAPPINGS

Once a light probe image is captured, it needs to be stored in an image file using an omnidirectional image mapping. This section describes four of the most commonly used image mappings and provides formulas to determine the appropriate (u, v) coordinates in the image corresponding to a unit direction in the world $D = (D_x, D_y, D_z)$, and vice versa. These formulas all assume a right-handed coordinate system in which $(0, 0, -1)$ is forward, $(1, 0, 0)$ is right, and $(0, 1, 0)$ is up.

This section also discusses some of the advantages and disadvantages of each format. These considerations include the complexity of the mapping equations, how much distortion the mapping introduces, and whether the mapping has special features that facilitate computing properties of the image or using it with specific rendering algorithms. The mappings this section presents are the *ideal mirrored sphere*, the *angular map*, *latitude–longitude*, and the *cube map*.

11.3.1 IDEAL MIRRORED SPHERE

For the ideal mirrored sphere, we use a circle within the square image domain of $u \in [0, 1], v \in [0, 1]$. The mapping equation for world to image is as follows.

$$r = \frac{\sin\left(\frac{1}{2}\arccos(-D_z)\right)}{2\sqrt{{D_x}^2 + {D_y}^2}}$$

$$(u, v) = \left(\frac{1}{2} + rD_x\ ,\ \frac{1}{2} - rD_y\right)$$

The mapping equation for image to world is as follows.

$$r = \sqrt{(2u-1)^2 + (2v-1)^2}$$

$$(\theta, \phi) = (\text{atan2}(2u-1, -2v+1), 2\arcsin(r))$$

$$(D_x, D_y, D_z) = (\sin\phi\cos\theta, \sin\phi\sin\theta, -\cos\phi)$$

The ideal mirrored sphere mapping (Figure 11.18[a]) is how the world looks when reflected in a mirrored sphere, assuming an orthographic camera and a world that

FIGURE 11.18 *(a) The ideal mirrored sphere mapping. (b) The angular map mapping. The mappings appear similar, but devote different amounts of image area to the front versus the back half of the environment.*

is distant relative to the diameter of the sphere. In practice, real spheres exhibit a mapping similar to this ideal one as long as the sphere is small relative to its distance to the camera and to the objects in the environment. Of course, the reflection in a sphere is actually a mirror image of the environment, and thus the image should be flipped in order to be consistent with directions in the world.

Like all mappings in this section, the ideal mirrored sphere reflects all directions of the environment. Straight forward — that is, $D = (0, 0, -1)$ — appears in the center of the image. Straight right, $D = (1, 0, 0)$, appears $\sqrt{2}/2$ of the way from the center to the right-hand edge of the image. Straight up appears $\sqrt{2}/2$ of the way from the center to the top of the image. Straight back is the one direction that does not cleanly map to a particular image coordinate, and corresponds to the outer circumference of the edge of the circle.

From these coordinates, we see that the front half of the environment is contained within a disk, that is, $\sqrt{2}/2$ of the diameter of the full image. This makes the area

taken up by the front half of the environment precisely equal to the area taken up by the back half of the environment. This property generalizes in that any two regions of equal solid angle in the scene will map to the same amount of area in the image. One use of this equal-area property is to calculate the average illumination color in a light probe (the average pixel value within the image circle is the average value in the environment). If the image has a black background behind the circular image, the average value in the scene is $4/\pi$ of the average of the entire square.

A significant disadvantage with the mirrored sphere mapping is that the back half of the environment becomes significantly stretched in one direction and squeezed in the other. This problem increases in significance toward the outer edge of the circle, becoming extreme at the edge. This can lead to the same problem we saw with mirrored sphere light probe images, in that the regions around the edge are poorly sampled in the radial direction. Because of this, the mirrored sphere format is not a preferred format for storing omnidirectional images, and the angular map format is frequently used instead.

11.3.2 ANGULAR MAP

For the angular map, we also use a circle within the square image domain of $u \in [0, 1]$, $v \in [0, 1]$. The mapping equation for world to image is as follows.

$$r = \frac{\arccos(-D_z)}{2\pi\sqrt{{D_x}^2 + {D_y}^2}}$$

$$(u, v) = \left(\frac{1}{2} - rD_y, \frac{1}{2} + rD_x\right)$$

The equation for image to world is as follows.

$$(\theta, \phi) = \left(\text{atan2}(-2v + 1, 2u - 1), \phi = \pi\sqrt{(2u - 1)^2 + (2v - 1)^2}\right)$$

$$(D_x, D_y, D_z) = (\sin\phi\cos\theta, \sin\phi\sin\theta, -\cos\phi)$$

The angular map format (Figure 11.18[b]) is similar in appearance to the mirrored sphere mapping, but it samples the directions in a manner that avoids undersampling the regions around the edges. In this mapping, the distance of a

point from the center of the image is directly proportional to the angle between straight ahead and its direction in the world. In this way, straight forward appears at the center of the image, and straight right and straight up appear halfway to the edge of the image. Regions that map near the edge of the sphere are sampled with at least as many pixels per degree in any direction as the center of the image. Because of this property, many light probe images (including those in the Light Probe Image Gallery [57]) are available in this format. Unlike the mirrored sphere mapping, the angular map is not equal-area and does not translate solid angle proportionately into image area. Areas near the edge become stretched in the direction tangent to the circumference but are neither stretched nor squeezed in the perpendicular direction, making them overrepresented in the mapping.

Because the mirrored sphere and angular map mappings appear similar, an angular map image is sometimes loaded into a rendering program as if it were a mirrored sphere image, and vice versa. The result is that the environment becomes distorted, and straight vertical lines appear curved. One way to tell which mapping such an image is in is to convert it to the latitude–longitude format (in which vertical lines should be straight) or the cube map format, in which all straight lines should be straight except at face boundaries.

11.3.3 LATITUDE–LONGITUDE

For the latitude–longitude mapping, we use a rectangular image domain of $u \in [0, 2]$, $v \in [0, 1]$. The mapping equation for world to image is as follows.

$$(u, v) = \left(1 + \frac{1}{\pi} \operatorname{atan2}(D_x, -D_z), \frac{1}{\pi} \arccos D_y\right)$$

The equation for image to world is as follows.

$$(\theta, \phi) = (\pi(u - 1), \pi v)$$
$$(D_x, D_y, D_z) = (\sin\phi \sin\theta, \cos\phi, -\sin\phi \cos\theta)$$

The latitude–longitude mapping (Figure 11.19[a]) maps a direction's azimuth to the horizontal coordinate and its elevation to the vertical coordinate of the image.

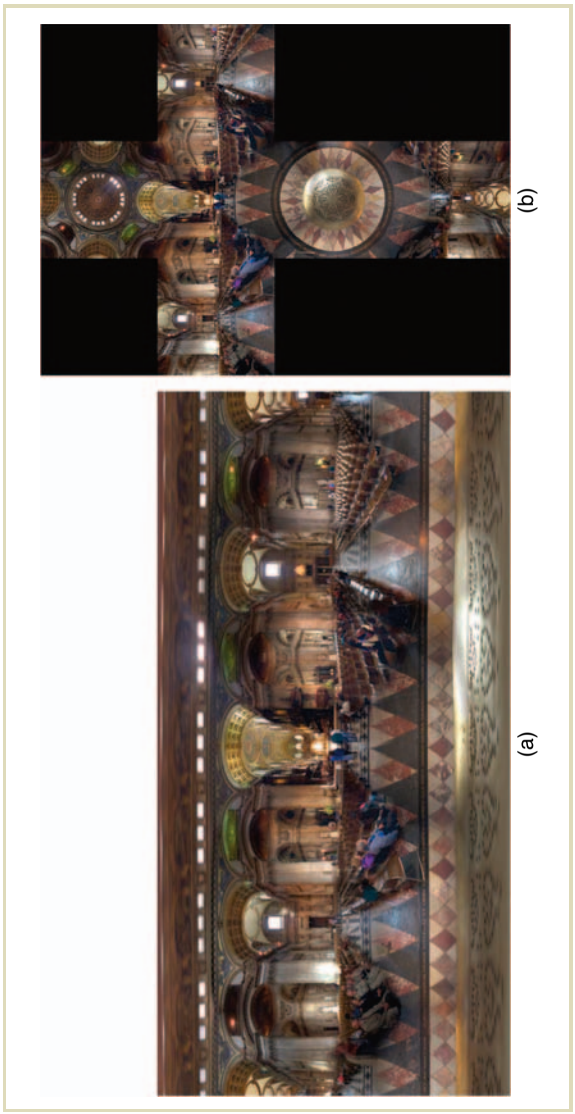


FIGURE 11.19 (a) The latitude–longitude mapping. (b) The cube map mapping.

This is known to cartographers as an "equirectangular mapping." Unlike the previous two mappings, it flattens the sphere into a rectangular area. The top edge of the image corresponds to straight up, and the bottom edge of the image is straight down. The format most naturally has a 2:1 aspect ratio (360° by 180°), as this introduces the least distortion for regions near the horizon. The areas toward straight up and straight down are sampled equivalently with the regions near the horizon in the vertical direction, and are progressively more oversampled toward the top and bottom edges.

The latitude–longitude mapping is convenient because it is rectangular and has no seams (other than at the poles), and because the mapping formulas are simple and intuitive. Although straight lines generally become curved in this format, vertical lines in the scene map to straight vertical lines in the mapping. Another useful property is that in this format, the lighting environment can be rotated around the y axis simply by translating the image horizontally. The mapping is not equal-area (the percentage any particular area is overrepresented is inversely proportional to the cosine of the latitude ϕ). Thus, to find the average pixel value in a light probe image in this format, one can multiply the image by the vertical cosine falloff function $\cos\phi$ and compute the average value of these modified pixels.

11.3.4 CUBE MAP

For the cube map, we use a rectangular image domain of $u \in [0, 3]$, $v \in [0, 4]$. The cube map formulas require branching to determine which face of the cube corresponds to each direction vector or image coordinate. Thus, they are presented as pseudocode. The code for world to image is as follows.

```
if ((Dz<0) && (Dz<=-abs(Dx))
           && (Dz<=-abs(Dy))) // forward
  u = 1.5 - 0.5 * Dx / Dz;
  v = 1.5 + 0.5 * Dy / Dz;
else if ((Dz>=0) && (Dz>=abs(Dx))
                 && (Dz>=abs(Dy))) // backward
```

```
  u = 1.5 + 0.5 * Dx / Dz;
  v = 3.5 + 0.5 * Dy / Dz;
else if ((Dy<=0) && (Dy<=-abs(Dx))
                 && (Dy<=-abs(Dz)) // down
  u = 1.5 - 0.5 * Dx / Dy;
  v = 2.5 - 0.5 * Dz / Dy;
else if ((Dy>=0) && (Dy>=abs(Dx))
                 && (Dy>=abs(Dz))) // up
  u = 1.5 + 0.5 * Dx / Dy;
  v = 0.5 - 0.5 * Dz / Dy;
else if ((Dx<=0) && (Dx<=-abs(Dy))
                 && (Dx<=-abs(Dz))) // left
  u = 0.5 + 0.5 * Dz / Dx;
  v = 1.5 + 0.5 * Dy / Dx;
else if ((Dx>=0) && (Dx>=abs(Dy))
                 && (Dx>=abs(Dz))) // right
  u = 2.5 + 0.5 * Dz / Dx;
  v = 1.5 - 0.5 * Dy / Dx;
```

The code for image to world is as follows.

```
if u>=1 and u<2 and v<1 // up
  Vx = (u - 1.5) * 2
  Vy = 1.0
  Vz = (v - 0.5) * -2
else if u<1 and v>=1 and v<2 // left
  Vx = -1.0
  Vy = (v - 1.5) * -2
  Vz = (u - 0.5) * -2
else if u>=1 and u<2 and v>=1 and v<2 // forward
  Vx = (u - 1.5) * 2
  Vy = (v - 1.5) * -2
  Vz = -1.0
else if u>=2 and v>=1 and v<2 // right
  Vx = 1.0
  Vy = (v - 1.5) * -2
  Vz = (u - 2.5) * 2
```

```
else if u>=1 and u<2 and v>=2 and v<3 // down
  Vx = (u - 1.5) * 2
  Vy = -1.0
  Vz = (v - 2.5) * 2
else if u>=1 and u<2 and v>=3 // backward
  Vx = (u - 1.5) * 2
  Vy = (v - 3.5) * 2
  Vz = 1.0

normalize = 1 / sqrt(Vx * Vx + Vy * Vy + Vz * Vz)

Dx = normalize * Vx
Dy = normalize * Vy
Dz = normalize * Vz
```

In the cube map format (Figure 11.19[b]), the scene is represented as six square perspective views, each with a 90° field of view, which is equivalent to projecting the environment onto a cube and then unfolding it. The six squares are most naturally unfolded into a horizontal or vertical cross shape but can also be packed into a 3×2 or 6×1 rectangle to conserve image space. The mapping is not equal-area (areas in the corners of the cube faces take up significantly more image area per solid angle than the areas in the center of each face). However, unlike the angular map and latitude–longitude mappings, this relative stretching is bounded: Angular areas that map to the cube's corners are overrepresented by a factor of up to $3\sqrt{3}$ in area relative to regions in the center.

This mapping requires six different formulas to convert between world directions and image coordinates, depending on which face of the cube the pixel falls within. Although the equations include branching, they can be more efficient to evaluate than the other mappings (which involve transcendental functions such as *asin* and *atan*2). This image format is sometimes the most convenient for editing the light probe image because straight lines in the environment remain straight in the image (although there are generally directional discontinuities at face boundaries). For showing a light probe image in the background of a real-time rendering application, the image mapping is straightforward to texture map onto a surrounding cubical surface.

11.4 HOW A GLOBAL ILLUMINATION RENDERER COMPUTES IBL IMAGES

Returning to the rendering process, it is instructive to look at how a global illumination algorithm computes IBL images such as those seen in Figure 11.4. In general, the algorithm needs to estimate how much light arrives from the lighting environment and the rest of the scene at each surface point, which in large part is a matter of visibility: Light from the visible parts of the environment must be summed, and light that is blocked by other parts of the scene must instead be replaced by an estimate of the light reflecting from those surfaces. This measure of incident illumination, multiplied by the reflectance of the surface itself, becomes the color rendered at a particular pixel in the image.

The RNL animation was rendered with RADIANCE [349], which like most modern global illumination systems is based on ray tracing. The image is rendered one pixel at a time, and for each pixel, the renderer needs to determine the RGB color L to display for that pixel. In our case, L is an HDR RGB pixel value, with its three components proportional to the amount of red, green, and blue radiance arriving toward the camera in the direction corresponding to the pixel. For each pixel, a ray R is traced from the camera C (Figure 11.20[a]) until it hits a surface in the scene at a 3D point P. L is then computed as a function of the reflectance properties of the surface at P and the incident light arriving at P. This section lists the different types of surfaces R can hit and how the rendering system then computes their appearance as lit by the scene. We will see that the most costly part of the process is computing how light reflects from diffuse surfaces. Later in this chapter, we will see how understanding this rendering process can motivate techniques for increasing the rendering efficiency.

Case 1: R Hits the IBL Environment If the ray strikes the emissive surface surrounding the scene at point P (Figure 11.20[a]), the pixel color L in the rendering is computed as the color from the light probe image that was texture mapped onto the surface at P. In RNL, this could be a green color from the leaves of the trees, a bright blue color from the sky, or a brown color from the ground below, depending on which pixel and where the camera is looking. The HDR range of the surrounding environment is transferred to the resulting HDR renderings.

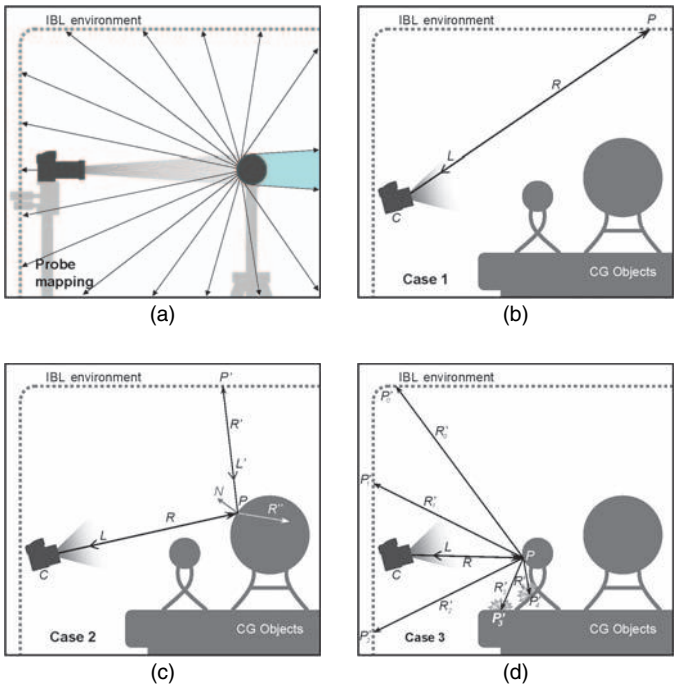


FIGURE 11.20 **Probe mapping(a):** *A mirrored sphere reflects the entire surrounding environment except for the light-blue area obscured by the ball. The ideal mirrored sphere mapping (see Section 11.3.1) maps a light probe image onto an IBL environment surface.* ***Case 1(b):*** *A ray R sent from the virtual camera C hits the IBL environment surface at P. The HDR value L on the surface is copied to the image pixel.* ***Case 2(c):*** *R hits a specular surface of a CG object. The reflected ray R' is traced, in this case striking the IBL environment at P' with an HDR value of L'. L' is multiplied by the specular color of the object to determine the final pixel value L. (For a translucent surface, a refracted ray R'' is also traced.)* ***Case 3(d):*** *R hits a diffuse surface at P. A multitude of rays R'_i are traced into the scene to determine the irradiance E at P as a weighted average of the incident light values L'_i from points P'_i. E is multiplied by the diffuse object color to determine the final pixel value L.*

In this case, the renderer does not take into consideration how the surface at point P is being illuminated because the surface is specified to be emissive rather than reflective. This is not exactly how light behaves in the real world, in that most objects placed in a scene have at least a minute effect on the light arriving at all other surfaces in the scene. In RNL, however, the only place where this might be noticeable is on the ground close to the pedestal, which might receive less light from the environment because of shadowing from the pedestal. Because this area was not seen in the RNL animation, the effect did not need to be simulated.

Case 2: R Hits a Mirror-like Specular Surface A mirror-like specular surface having no roughness shows a clear image of the scene around it. In RNL, the glass, plastic, and metallic spheres have mirror-like specular components. When the ray R hits such a surface at a point P (Figure 11.20[b]), the renderer reflects the ray about P's surface normal N and follows this reflected ray R' into the scene until it strikes a new surface at point P'. It then recursively computes the color L' of the light coming from P' along R' toward P in precisely the same way it computes the light coming from a point P along R toward the camera. For example, if R' strikes the emissive surface surrounding the scene at P', the renderer retrieves the appropriate pixel color L' from the surrounding light probe image (an instance of Case 1). The recursion depth for computing bounces of light is usually limited to a user-defined number, such as six bounces for specular reflections and two bounces for diffuse reflections.

The incident light L' along the reflected ray is then used to produce the specular component of the resulting light L reflecting from the surface. For metals, this light is multiplied by the metallic color of the surface. For example, a gold material might have a metallic color of (0.8, 0.6, 0.3). Because these color components are less than 1, the metallic reflection will reveal some of the detail in the reflected HDR environment not seen directly.

For glass-like materials, the mirror-like specular component is fainter, typically in the range of 4–8% (although at grazing angles it becomes greater as a result of Fresnel reflection), depending on the index of refraction of the material. Thus, the light L' is multiplied by a small value to create the specular component of the color L in the image. In the case of RNL, this makes the bright sky detail particularly evident in the reflections seen in the top of the large glass ball (as in Figure 11.4[a]). For glass-like materials, a second refracted ray R'' is also traced through the translucent

surface, and the light L'' arriving along this ray is added to the total light L reflected toward the camera.

For plastic materials, there is both a specular component and a diffuse component to the reflection. For such materials, the specular component is computed in the same way as the specular component of glass-like materials and is added to the diffuse component, which is computed as described in material following.

Case 3: R Hits a Diffuse Surface Diffuse surfaces reflect light equally in all directions, and light arriving from every direction in the upper hemisphere of the surface contributes to the reflected light. Because of this, computing the light reflecting from diffuse surfaces can be computationally expensive. The total amount of light arriving at a surface point P is called its "irradiance," denoted by E, which is a weighted integral of all colors L'_i arriving along all rays R'_i (Figure 11.20[c]). The contribution of each ray R'_i is weighted by the cosine of the angle θ_i between it and the surface normal N because light arriving from oblique angles provides less illumination per unit area. If we denote $L'(P, \omega)$ to be the function representing the incident light arriving at P from the angular direction ω in P's upper hemisphere Ω, E can be written as

$$E(P, N) = \int_{\Omega} L'(P, \omega) \cos\theta \, d\omega.$$

Unfortunately, E cannot be computed analytically because it is dependent not only just on the point's view of the environment but also on how this light is occluded and reflected by all other surfaces in the scene visible to the point. To estimate E, the renderer takes a weighted average of the light colors L'_i arriving from a multitude of rays R'_i sent out from P to sample the incident illumination. When a ray R'_i strikes the surface surrounding the scene, it adds the corresponding pixel color L'_i from the lighting environment to the sum, weighted by the cosine of the angle between N and R'_i. When a ray R'_i strikes another part of the scene P'_i, the renderer recursively computes the color of light L'_i reflected from P'_i toward P and adds this to the sum as well, again weighted by $\cos\theta$. Finally, E is estimated as this sum is divided by the total number of rays sent out, as follows.

$$E(P, N) \approx \frac{1}{k} \sum_{i=0}^{k-1} L'_i \cos\theta_i$$

The accuracy of the estimate of E increases with the number of rays k sent out from P. Once E is computed, the final light L drawn for the pixel is the surface's diffuse color (called its "albedo," often denoted by ρ) multiplied by the irradiance E.

Computing the integral of the incident illumination on the surface of an object performs a blurring process on the HDR light probe image that is similar to the blurring we have seen in Section 11.1.5. HDR pixels in the image are averaged together according to a filter, in this case a blur over the upper hemisphere. Just as before, this process makes the effect of HDR pixel values in an environment visible in the much lower dynamic range values reflecting from diffuse surfaces. Clipping the lighting environment to the white point of a display before computing the lighting integral would significantly change the values of the computed irradiance.

Because many rays must be traced, the process of sampling the light arriving from the rest of the scene at a point can be computationally intensive. When rays are sent out at random, the number of rays needed to estimate the incident illumination to a given expected accuracy is proportional to the variance of the illumination within the scene. To conserve computation, some renderers (such as RADIANCE) can compute E for a subset of the points P in the scene, and for other points interpolate the irradiance from nearby samples — a process known as "irradiance caching." Furthermore, irradiance samples computed for one frame of an animation can be reused to render subsequent frames, as long as the scene remains static. Both of these features were used to reduce the rendering time for the RNL animation by a factor of several thousand.

11.4.1 SAMPLING OTHER SURFACE REFLECTANCE TYPES

In RNL, all of the surfaces had either a mirror-like specular component or completely diffuse reflectance, or a combination of the two. Other common material types include rough specular reflections and more general bidirectional reflectance distribution functions (BRDFs) [239] that exhibit behaviors such as retroreflection and anisotropy. For such surfaces, the incident illumination needs to be sampled according to these more general distributions. For example, in the case of rough specular surfaces, the rendering system needs to send out a multitude of rays in the general direction of the reflected angle R' with a distribution whose spread varies with the specular roughness parameter. Several BRDF

models (such as Lafortune et al. [170] and Ashikhmin and Shirley [13]) have associated sampling algorithms that generate reflected ray directions in a distribution that matches the relative contribution of incident light directions for any particular viewing direction. Having such *importance sampling* functions is very helpful for computing renderings efficiently, as the alternative is to send out significantly more rays with a uniform distribution and weight the incident light arriving along each ray according to the BRDF. When the number of rays is limited, this can lead to noisy renderings in comparison with sampling in a distribution that matches the BRDF. Figure 11.21 shows an example of this difference from Lawrence et al. [178].

In general, using more samples produces higher-quality renderings with more accurate lighting and less visible noise. We have just seen that for different types of

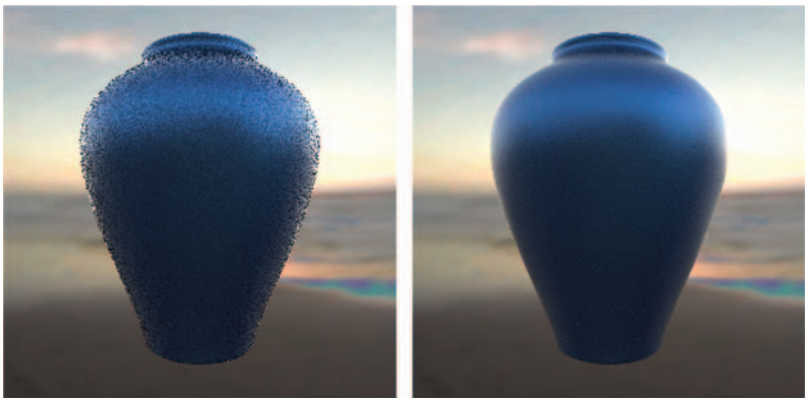


FIGURE 11.21 *IBL renderings of a pot from [178] computed with 75 rays per pixel to sample the incident illumination for a vase with a rough specular BRDF. (a) Noisy results obtained by using uniform sampling and modulating by the BRDF. (b) A result with less noise obtained by sampling with a distribution based on a factored representation of the BRDF.*

materials, renderings are created most efficiently when samples are chosen according to a distribution that matches the relative importance of each ray to the final appearance of each pixel. Not surprisingly, the importance of different ray directions depends not only on the BRDF of the material but also on the distribution of the incident illumination in the environment. To render images as efficiently as possible, it becomes important to account for the distribution of the incident illumination in the sampling process, particularly for IBL environments with bright concentrated light sources. The next section presents this problem and describes solutions for sampling incident illumination efficiently.

11.5 SAMPLING INCIDENT ILLUMINATION EFFICIENTLY

We have just seen that the speed at which images can be computed using IBL as described in the previous section depends on the number of rays that need to be traced. The bulk of these rays are traced to estimate the illumination falling on diffuse surfaces through sampling. The light falling on a surface (i.e., its irradiance E) is the average value of the radiance arriving along all light rays striking the surface, weighted by the cosine of the angle each ray makes with the surface normal. Because averaging together the light from every possible ray is impractical, global illumination algorithms estimate the average color using just a finite sampling of rays. This works very well when the lighting environment is generally uniform in color. In this case, the average radiance from any small set of rays will be close to the average from all of the rays because no particular ray strays far from the average value to begin with. However, when lighting environments have some directions that are much brighter than the average color, it is possible for the average of just a sampling of the rays to differ greatly from the true average.

When ray directions are chosen at random, the number of rays needed to accurately sample the illumination is proportional to the variance in the light probe image. Lighting environments that have low variance (such as cloudy skies) can be sampled accurately with just tens of rays per irradiance calculation. Environments with greater variance, such as scenes with concentrated light sources, can require hundreds or thousands of rays per irradiance calculation when rays are sent out at random. The eucalyptus grove light probe, which featured bright backlit clouds in the direction of the setting sun, required over 1000 rays per irradiance calculation,

FIGURE 11.22 *A laser-scanned 3D model of the Parthenon illuminated by a uniformly white lighting environment, giving the appearance of a dense cloudy day.*

making rendering RNL computationally expensive. In this section, we describe this sampling problem in detail and present several sampling techniques that mitigate the difficulties.

Figure 11.22 shows a virtual model of the Parthenon rendered with IBL using a synthetic lighting environment of a completely white sky.[6] This environment, being all the same color, has zero variance, and a high-quality rendering could be computed using the Arnold global illumination rendering system [85] using a relatively modest 81 rays to sample the incident light at each pixel. The only source of illumination variance at each surface point is because of visibility: Some directions see out to the sky, whereas others see indirect light from the other surfaces in the scene. Because the color of the indirect light from these other surfaces is also low in variance and not radically different from the light arriving from the sky, the

6 The virtual Parthenon model was created by laser scanning the monument and using an inverse global illumination process leveraging IBL to solve for its surface colors and reflectance properties from photographs, as described in Debevec et al. [60].

FIGURE 11.23 *A light probe image of a sunny sky, taken with a fish-eye lens. Pixels on the sun disk (much smaller than the saturated region seen here) are on the order of 100,000 times brighter than the average color of the rest of the sky.*

total variance remains modest and the illumination is accurately sampled using a relatively small number of rays.

In contrast to the perfectly even sky, Figure 11.23 shows a clear sky with a directly visible sun, acquired using the technique in Stumpfel et al. [310]. The half-degree disk of the sun contains more than half the sky's illumination but takes up only one hundred thousandth of the area of the sky, giving the sunny sky a very high variance. If we use such a high-variance environment to illuminate the Parthenon model, we obtain the noisy rendering seen in Figure 11.24. For most of the pixels, none of the 81 rays sent out to sample the illumination hit the small disk of the sun, and thus they appear to be lit only by the light from the sky and clouds. For the pixels where one of the rays did hit the sun, the sun's contribution is greatly overestimated because the sun is counted as 1/81 of the lighting environment rather than one hundred thousandth. These pixels are far brighter than what can be shown in an

FIGURE 11.24 *The virtual Parthenon rendered using the sunny sky light probe in Figure 11.23 with 81 lighting samples per pixel. The appearance is speckled because 81 samples distributed randomly throughout the sky are not enough to accurately sample the small disk of the sun.*

LDR image, and their intensity is better revealed by blurring the image somewhat (as in Figure 11.25). On an average, this image shows the correct lighting, but the appearance is strange because the light from the sun has been squeezed into a random scattering of overly bright pixels.

The noisy Parthenon rendering shows what is known as the "sampling problem." For lighting environments with concentrated sources, a very large number of rays needs to be sent in order to avoid noisy renderings. For the sunny sky, hundreds of thousands of rays would need to be sent to reliably sample the small sun and the large sky, which is computationally impractical. Fortunately, the number of rays can be reduced to a manageable number using several techniques. In each technique, the key idea is to give the rendering algorithm *a priori* information about how to sample

FIGURE 11.25 *A Gaussian blurred version of Figure 11.24, showing more clearly the amount of reflected sun energy contained in just a few of the image pixels.*

the illumination environment efficiently. These techniques (described in material following) include *light source identification*, *light source constellations*, and *importance sampling*.

11.5.1 IDENTIFYING LIGHT SOURCES

The ray-sampling machinery implemented in traditional global illumination algorithms typically samples only the indirect lighting within a scene. Concentrated light sources are assumed to be explicitly modeled and taken into account in the direct lighting calculation in which rays are sent to these lights explicitly. In IBL, both direct and indirect sources are effectively treated as indirect illumination, which strains the effectiveness of this sampling machinery.

Many IBL environments can be partitioned into two types of areas: small concentrated light sources and large areas of indirect illumination. These types of environments fare poorly with simplistic sampling algorithms, in that much of the illumination is concentrated in small regions that are easily missed by randomly sampled rays. One method of avoiding this problem is to identify these small concentrated light regions and convert them into traditional area light sources. These new area light sources should have the same shape, direction, color, and intensity as seen in the original image, and the corresponding bright regions in the IBL environment must be removed from consideration in the sampled lighting computation. This yields the type of scene that traditional global illumination algorithms are designed to sample effectively.

For the sunny sky light probe, this process is straightforward. The direction of the sun can be determined as the centroid of the brightest pixels in the image, and converted to a world direction vector using the angular mapping formula in Section 11.3.2. As mentioned earlier in this chapter, the size and shape of the sun is known to be a disk whose diameter subtends 0.53° of the sky. The color and intensity of the sun can be obtained from the light probe image as the average RGB pixel value of the region covered by the sun disk. In a typical rendering system, a specification for such a light source might look as follows.

```
light sun directional {
  direction -0.711743 -0.580805 -0.395078
  angle 0.532300
  color 10960000 10280000 866000
}
```

Figure 11.26 shows a global illumination rendering of the Parthenon illuminated just by the sun light source. For each pixel, the renderer explicitly sends at least one ray toward the disk of the sun to sample the sun's light because the sun is a direct light source known *a priori* to the renderer. Thus, the rendering has no noise problems, as in Figure 11.24. Although the rest of the sky is black, the renderer still sends additional randomly fired rays from each surface to estimate the indirect illumination arriving from other surfaces in the scene. These effects are most significant in the case of shadowed surfaces that are visible to sunlit surfaces, such as the left sides of the front columns. Because the blue skylight scattered by the atmosphere is

FIGURE 11.26 *The Parthenon illuminated only by the sun, simulated as a direct light source.*

not being simulated, the rendering shows how the Parthenon might look if it were located on the moon and illuminated by a somewhat yellowish sun.

The rest of the sky's illumination can be simulated using an IBL process, but we first need to make sure that the light from the sun is no longer considered to be part of the IBL environment. In some rendering systems, it is sufficient to place the new direct light source in front of the IBL environment surface, and it will occlude the image-based version of the source from being hit by indirect sample rays. In others, the corresponding image region should be set to black in order to prevent it from being part of the image-based illumination. If we remove the sun from the sky and use the remainder of the image as an IBL environment, we obtain the rendering seen in Figure 11.27. This rendering, although it lacks sunlight, is still a realistic one. It shows the scene approximately as if a cloud had passed in front of the sun.

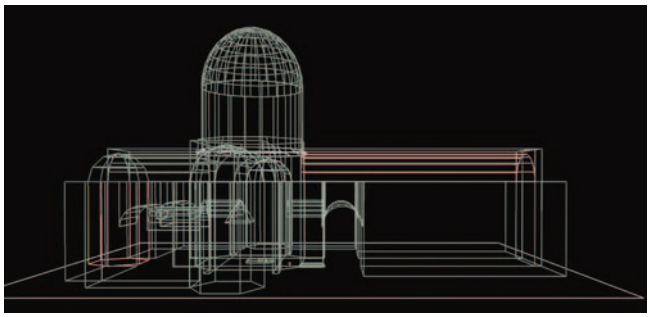
FIGURE 11.27 *The Parthenon illuminated only by the sky and clouds, with 81 samples per pixel.*

In practice, it is sometimes necessary to delete a somewhat larger area around the light source from the IBL environment because some light sources have significantly bright regions immediately near them. These regions on their own can contribute to the appearance of noise in renderings. Sometimes, these regions are because of glare effects from imperfect camera optics. In the case of the sun, forward-scattering effects in the atmosphere create a bright circumsolar region around the sun. Because of both of these effects, to create the sky used for Figure 11.27, a region covering the circumsolar area was removed from the light probe image, and this additional light energy was added to the sun intensity used to render Figure 11.26. As a result, the edges of shadows in the sun rendering are slightly sharper than they should be, but the effect is a very subtle one.

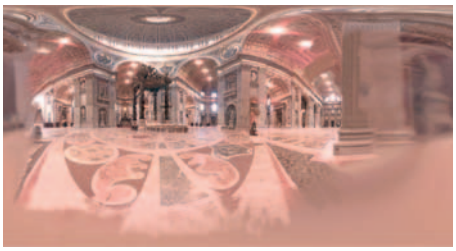
FIGURE 11.28 *An efficient noise-free rendering of the Parthenon made using IBL for the sky and clouds and a direct light source for the sun, with 81 samples per pixel.*

Finally, we can light the scene with the sun and sky by simultaneously including the IBL environment for the sky and clouds and the direct light source for the sun in the same rendering, as seen in Figure 11.28. (We could also just add the images from Figures 11.26 and 11.27 together.) This final rendering shows the combined illumination from the sun and the rest of the sky, computed with a relatively modest 81 sample rays per pixel.

Identifying light sources can also be performed for more complex lighting environments. The SIGGRAPH 99 Electronic Theater animation *Fiat Lux* used an IBL environment created from HDR images acquired in St. Peter's Basilica. It was rendered with the RADIANCE lighting simulation system. The images were assembled into HDR panoramas and projected onto the walls of a basic 3D model of the Basilica's interior, seen in Figure 11.29(b). The illumination within the Basilica consisted of



(a)



(b)

(c)

FIGURE 11.29 (a) Identified light sources from (c) corresponding to the windows and the incandescent lights in the vaulting. The HDR image and the light sources were projected onto a 3D model of the Basilica interior to form a 3D lighting environment. (b) Basic 3D geometry of the Basilica interior used as the IBL environment surface. (c) An IBL environment acquired inside St. Peter's Basilica.

indirect light from the walls and ceiling, as well as concentrated illumination from the windows and the incandescent lights in the vaulting. To create renderings efficiently, each of the concentrated area lights' sources was identified by drawing a polygon around the source in the panoramic image, as seen in Figure 11.29(a). A simple program computed the average HDR pixel value of the region covered by each light source, and created an area light source set to this value for its radiance. Just like the panoramic image, the vertices of the identified light sources were projected onto the walls of the virtual Basilica model, placing the lights at their proper 3D locations within the scene. Thus, the image-based illumination changed throughout the virtual scene, as the directional light depended on each object's 3D location relative to the light sources.

Two other details of the procedure used in *Fiat Lux* are worth mentioning. First, the light sources were placed at a small distance in front of the walls of the Basilica, so that rays fired out from the surfaces would have no chance of hitting the bright regions behind the lights without having to set these regions to black. Second, the lights were specified to be "illum" light sources, a special RADIANCE light source type that is invisible to rays coming directly from the camera or from mirror-like reflections. As a result, the lights and windows appeared with their proper image-based detail when viewed directly and when seen in the reflections of the synthetic objects, even though they had been covered up by direct light sources. Figure 11.30(a) shows a frame from *Fiat Lux* in which a variety of virtual objects have been placed within the Basilica using IBL with identified light sources. Because "illum" sources were used, the motion-blurred reflections in Figure 11.30(b) reflect the original image-based light sources.

Identifying light sources dramatically reduces the number of rays needed to create noise-free renderings of a scene, and it maintains the realism and accuracy of IBL. Having the concentrated sources converted to individual CG lights is also useful for art direction, as these light sources can be readily repositioned and changed in their color and brightness. These light sources can also be used in a rendering system that does not support global illumination, yielding at least an approximate version of the IBL environment. For some applications, however, it is desirable to have a fully automatic method of processing an IBL environment into a description that can be rendered efficiently. The remainder of this section presents some of these automatic sampling techniques.

(a)

(b)

FIGURE 11.30 (a) A frame from the Fiat Lux animation, showing synthetic objects inserted into the Basilica, illuminated by the HDR lighting environment using identified light sources. (b) Another frame from the animation showing HDR motion blur effects in the reflections of the spinning objects. Shadows and reflections of the objects in the floor were created using the techniques described in Section 11.6.

11.5.2 CONVERTING A LIGHT PROBE INTO A CONSTELLATION OF LIGHT SOURCES

As we saw earlier, it is possible to reduce the variance in an IBL environment by converting concentrated spots of illumination into direct light sources. We can carry this idea further by turning *entire* lighting environments into constellations of light sources. These approximations can eliminate noise from renderings but can introduce aliasing in shadows and highlights when not enough light sources are used. When the light source directions are chosen with care, accurate illumination can be created using a manageable number of light sources, and these lights can be used either in traditional or global illumination rendering systems.

In general, this approach involves dividing a light probe image into a number of regions, and then creating a light source corresponding to the direction, size, color, and intensity of the total light coming from each region. Each region can be represented either by a point light source, or by an area light source corresponding to the size and/or shape of the region. Figure 11.31 shows perhaps the simplest case of approximating an IBL environment with a constellation of point light sources. A light probe taken within St. Peter's Basilica was converted to a cube map (Figure 11.31[a]), and each face of the cube map was resized to become a square of just 10×10 pixels, seen in Figure 11.31(a). Then, a point light source was placed in the direction corresponding to each pixel on each face of the cube, and set to the color and intensity of the corresponding pixel, yielding 600 light sources. These light sources produced a low-resolution point-sampled version of the IBL environment. The technique has the attractive quality that there is no sampling noise in the renderings, as rays are always traced to the same light locations. However, the technique can introduce *aliasing* because the finite number of lights may become visible as stair-stepped shadows and fragmented specular reflections.

Figure 11.31(b) shows the results of rendering a small scene using this constellation of light sources. For both the diffuse figure and the glossy red ball, the rendering is free of noise and artifacts, although the shadows and highlights from the windows and lights are not in precisely the right locations because of the finite resolution of the light source constellation. For the shiny sphere, simulating the illumination from the set of point lights makes little sense because there is a vanishingly small probability that any particular reflected ray would precisely hit one of the point lights. Instead, it makes sense to use ray tracing to reflect these rays directly

(a)

(b)

FIGURE 11.31 (a) A light probe image taken inside St. Peter's Basilica in cube map format. (b) An approximation of the St. Peter's lighting using a 10×10 array of point lights for each cube face. The Buddha model is courtesy of the Stanford computer graphics laboratory.

into the IBL environment, as described in Case 2 in Section 11.4. In the rendering, the mirrored sphere is shown reflecting an illustrative image composed of spots for each point light source.

The quality of approximating an IBL environment with a finite number of lights can be significantly increased if the light sources are chosen in a manner that conforms to the distribution of the illumination within the light probe image. One strategy for this is to have each light source represent approximately the same quantity of light in the image. Taking inspiration from Heckbert's median-cut color quantization algorithm [123], we can partition a light probe image in the rectangular latitude–longitude format into 2^n regions of similar light energy as follows.

1. Add the entire light probe image to the region list as a single region.
2. For each region, subdivide along the longest dimension such that its light energy is divided evenly.
3. If the number of iterations is less than n, return to step 2.

For efficiency, calculating the total energy within regions of the image can be accelerated using summed area tables [48]. Once the regions are selected, a light source can be placed in the center of each region, or alternately at energy centroid of each region, to better approximate the spatial distribution of the light within the region. Figure 11.32 shows the Grace Cathedral lighting environment partitioned into 256 light sources, and Figure 11.33 shows a small scene rendered with 16, 64, and 256 light sources chosen in this manner. Applying this technique to our simple diffuse scene, 64 lights produce a close approximation to a well-sampled and computationally intensive global illumination solution, and the 256-light approximation is nearly indistinguishable.

A few implementation details should be mentioned. First, computing the total light energy is most naturally performed on monochrome pixel values rather than RGB colors. Such an image can be formed by adding together the color channels of the light probe image, optionally weighting them in relation to the human eye's sensitivity to each color channel.[7] Partitioning decisions are made on this monochrome image, and light source colors are computed using the corresponding regions in the

7 Following ITU-R BT.709, the formula used to convert RGB color to monochrome luminance is $Y = 0.2125R + 0.7154G + 0.0721B$.

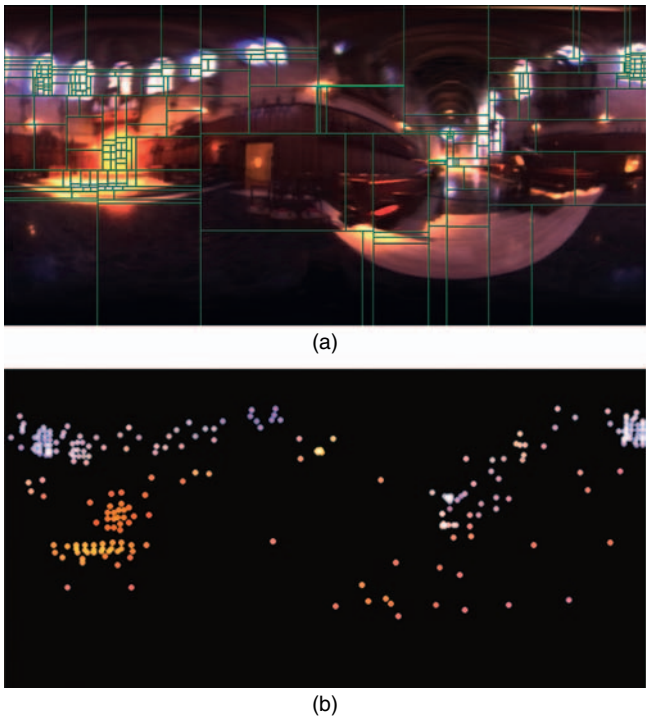


FIGURE 11.32 (a) The Grace Cathedral light probe image subdivided into 256 regions of equal light energy using the median-cut algorithm. (b) The 256 light sources chosen as the energy centroids of each region. All of the lights are approximately equal in intensity. A rendering made using these lights appears in Figure 11.33(c).

original color image. Second, the latitude–longitude format overrepresents the area of regions near the poles. To compensate for this, the pixels of the probe image should first be scaled by $\cos\phi$. Additionally, determining the longest dimension of a region should also take this stretching into account. This can be approximated by

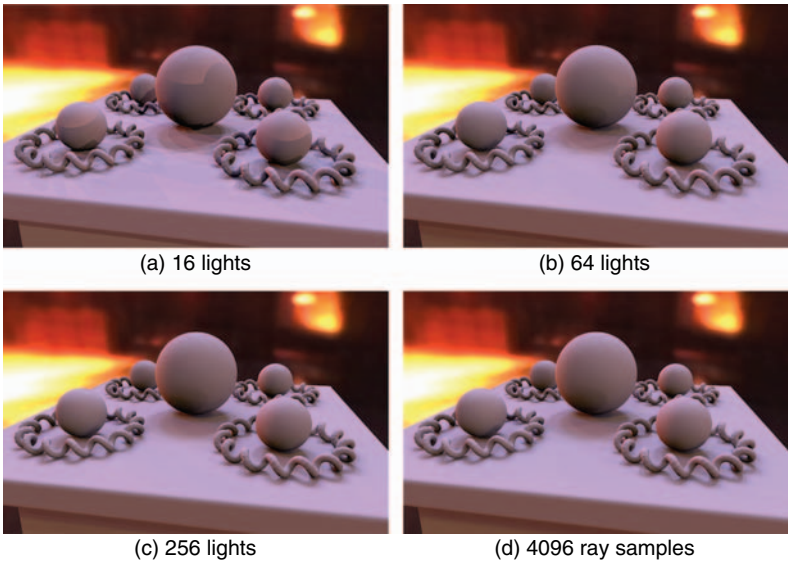


FIGURE 11.33 *Noise-free renderings (a through c) in the Grace Cathedral lighting environment as approximated by 16, 64, and 256 light sources chosen with the median cut algorithm. An almost noise-free Monte Carlo IBL rendering (d) needing 4096 randomly chosen rays per pixel.*

multiplying a region's width by $\cos\phi$, where ϕ is taken to be the angle of inclination of the middle of the region.

Several other light source selection procedures have been proposed for approximating light probe images as constellations of light sources. In each, choosing these point light source positions is also done with a clustering algorithm.

The LightGen plug-in [45] for HDR Shop [317] takes a light probe image in latitude–longitude format and outputs a user-specified number of point light sources in a variety of 3D modeling program formats. LightGen chooses its light source positions using a K-means clustering process [200]. In this process, K light source positions are initially chosen at random. Then, the pixels in the light

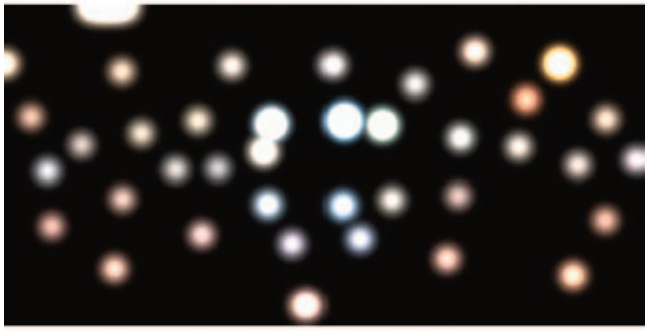
probe image are partitioned into sets according to which light source they are closest to. For each set of pixels, the mass centroid of the set is determined such that the mass of a pixel is proportional to its total intensity. Then, each of the K light sources is moved to the mass centroid of its set. The pixels are repartitioned according to the new light source directions and the process is repeated until convergence. Finally, each light source is assigned the total energy of all pixels within its set. Figure 11.34 shows results for $K = 40$ and $K = 100$ light sources for a kitchen light probe image.

LightGen tends to cluster the light sources around bright regions, but these light sources generally contain more energy than the light sources placed in dimmer regions. As a result, dimmer regions receive more samples than they do in the median-cut algorithm, but more lights may be necessary to approximate the structure of shadows. For example, if the kitchen window is approximated as six bright point light sources, it can be possible to observe multiple distinct shadows from the six sources rather than the expected soft shadow from an area light source.

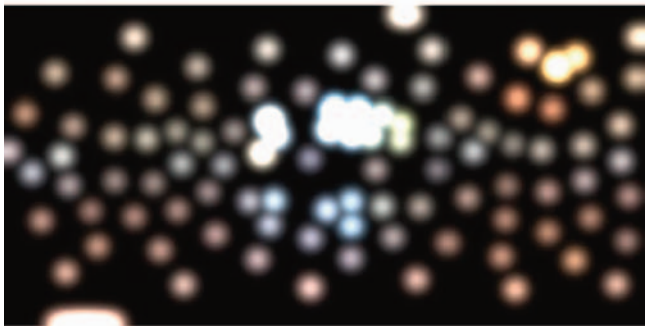
Kollig and Keller [160] propose several improvements to LightGen's clustering technique. They begin the K-means procedure using a single randomly placed light source and then add in one more random light at a time, each time iterating the K-means clustering procedure until convergence. This process requires additional computation but performs better at placing the light sources within concentrated areas of illumination. They also discuss several procedures of reducing aliasing once the light source positions are chosen. One of them is to use the light source regions as a structure for *stratified sampling*. In this process, the lighting environment is sampled with K rays, with one ray chosen to fire at random into each light source region. To avoid problems with the remaining variance within each region, they propose using the average RGB color of each region as the color of any ray fired into the region. Lights chosen by this algorithm and a rendering using such lights are shown in Figures 11.35(a) and (b).

FIGURE 11.34 *(a) An IBL environment of a kitchen. (b) An approximation of the kitchen environment made by LightGen using 40 point lights. (c) An approximation using 100 point lights.*

(a)



(b)



(c)

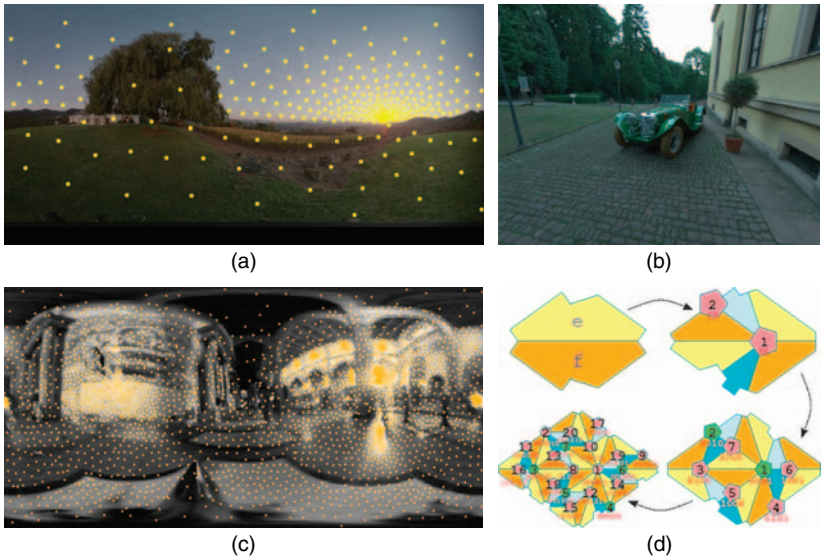


FIGURE 11.35 (a) Lights chosen for an outdoor light probe image by the Kollig and Keller sampling algorithm. (b) An IBL rendering created using lights from the Kollig and Keller algorithm. (c) Lights chosen for the Galileo's tomb light probe image by the Ostromoukhov et al. sampling algorithm. (d) The Penrose tiling pattern used by Ostromoukhov et al. for choosing light source positions.

Ostromoukhov et al. [242] used a light-source sampling pattern based on the geometric Penrose tiling to quickly generate a hierarchical sampling pattern with an appropriately spaced distribution. The technique is optimized for efficient sampling and was able to achieve sampling speeds of just a few milliseconds to generate several hundred light samples representing a lighting environment. A set of lights chosen using this pattern is shown in Figure 11.35(c), and the Penrose tiling pattern used is shown in Figure 11.35(d).

11.5.3 IMPORTANCE SAMPLING

As an alternative to converting a light probe image into light sources, one can construct a randomized sampling technique such that the rays are sent in a distribution that matches the distribution of energy in the light probe image. In the case of the sunny sky in Figure 11.23, such an algorithm would send rays toward the disk of the sun over half the time, since more than half of the light comes from the sun, rather than in proportion to the sun's tiny area within the image. This form of sampling can be performed using a mathematical technique known as "importance sampling."

Importance sampling was introduced for the purpose of efficiently evaluating integrals using Monte Carlo techniques [218] and has since been used in a variety of ways to increase the efficiency of global illumination rendering (e.g., Veach and Guibas [332]). Because a random number generator of a computer produces uniformly distributed samples, a process is needed to transform uniformly distributed numbers to follow the probability distribution function (PDF) corresponding to the distribution of light within the image. The process is most easily described in the context of a 1D function $f(x) : x \in [a, b]$, as seen in Figure 11.36(a). Based on a desired PDF, one computes the cumulative distribution function (CDF) $g(x) = \int_a^x f(x) / \int_a^b f(x)$, as seen in Figure 11.36(b). We note that $g(x)$ increases monotonically (and thus has an inverse) because $f(x)$ is nonnegative, and that $g(x)$ ranges between 0 and 1. Using $g(x)$, we can choose samples x_i in a manner corresponding to the distribution of energy in $f(x)$ by choosing values y_i uniformly distributed in $[0, 1]$ and letting $x_i = g^{-1}(y_i)$. This process is shown graphically for four samples in Figure 11.36(b).

To see why this process works, note that each bright light source near a point x in the PDF produces a quick vertical jump in the CDF graph in the area of $g(x)$. Seen from the side, the jump produces a flat span whose length is proportional to the amount of the scene's energy contained by the light. This flat span becomes a likely landing spot for randomly chosen samples y_i, with a likelihood proportional to the light's energy within the scene. When a sample y_i falls within the span, it produces a sample of the light source near x. In Figure 11.36(b), we see that three of the four random samples fell on spans corresponding to areas within bright light sources. The sample furthest to the top right did not land on a steep slope, and thus produced a sample in one of the dimmer areas of the image.

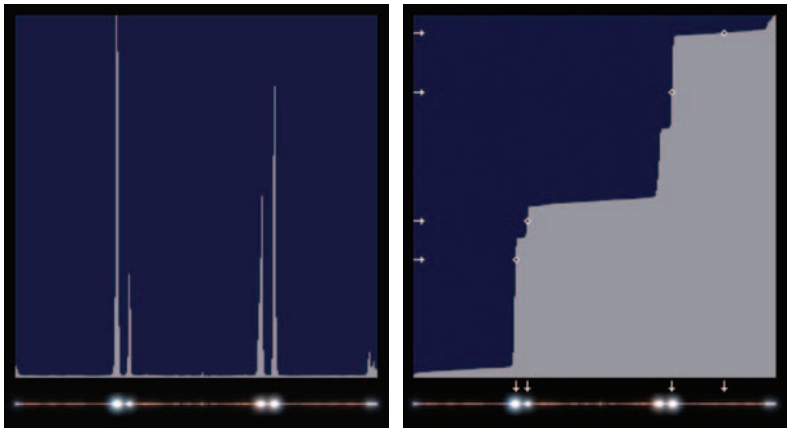


FIGURE 11.36 *Importance sampling. (a) A plot of the brightness of a 256×1 pixel region (seen below) of the St. Peter's light probe image that intersects several bright windows, forming a PDF. The image region is displayed below the graph using an HDR glare effect. (b) A graph of the CDF of the region from left to right for importance sampling. Four randomly chosen samples y_i (indicated by small horizontal arrows) are followed right until they intersect the graph (indicated by the diamonds) and are then followed down to their corresponding image pixel samples x_i.*

Extending this technique from 1D functions to 2D images is straightforward, as one can concatenate each row of an $m \times n$ light probe image into a single $mn \times 1$ pixel vector, suggested by Agarwal et al. [5]. Pharr and Humphreys [252] propose an implementation of importance sampling for light probe images within their physically based rendering package [253], in which importance sampling is first used to compute which column of the image to sample (the energy of each column x is summed to produce $f(x)$) and a sample is taken from the chosen image column using 1D importance sampling (as described earlier). Figure 11.37 compares the results of using this sampling algorithm to using randomly distributed

FIGURE 11.37 (a) A noisy IBL rendering in the St. Peter's lighting environment (Figure 11.31[a]) using 64 randomly distributed samples per pixel. (b) A much smoother rendering using 64 samples per pixel chosen using the importance sampling technique described in Pharr et al. [252].

rays. As desired, the rendering using importance sampling yields dramatically less noise than random sampling for the same number of samples.

To produce renderings that converge to the correct solution, light values chosen using importance sampling must be weighted in a manner that is inversely proportional to the degree of preference given to them. For example, sampling the sunny sky of Figure 11.23 with importance sampling would on an average send over half of the rays toward the sun, as the sun contains over half the light in the image. If we weighted the radiance arriving from all sampled rays equally, the surface point would be illuminated as if the sun were the size of half the entire sky.

A deficiency of these light probe sampling techniques is that they sample the lighting environment without regard to which parts of the environment are visible to a given surface point. Agarwal et al. [5] present a *structured importance sampling* approach for IBL that combines importance sampling with the conversion to light

sources in order to better anticipate variable visibility to the environment. They first threshold the image into regions of similar intensity, and use the Hochbaum–Shmoys clustering algorithm to subdivide each region a number of times proportional to both its size and its total energy. In this way, large dim regions do not become undersampled, which improves the rendering quality of shadowed areas. Likewise, small bright regions are represented with fewer samples relative to their intensity, in that their small spatial extent allows them to be accurately modeled with a relatively small number of samples. Cohen [44] constructs a piecewise-constant importance function for a light probe image and introduces a *visibility cache* that exploits coherence in which parts of the lighting environment are visible to neighboring pixels. With this cache, significantly fewer rays are traced in directions that are occluded by the environment.

11.6 SIMULATING SHADOWS AND SCENE–OBJECT INTERREFLECTION

So far, the IBL techniques presented the simulation of how the light from an environment illuminates CG objects, but not how the objects in turn affect the appearance of the environment. The most notable effect to be simulated is the shadow an object casts beneath itself. However, shadows are just part of the lighting interaction that should be simulated. Real objects also reflect light back onto the scene around them. For example, a red ball on the floor will make the floor around it somewhat redder. Effects such as this are often noticed only subconsciously but can contribute significantly to the realism of rendered images.

The examples we have seen so far from RNL and the Parthenon did not need to simulate shadows back into the scene. The Parthenon model was a complete environment with its own surrounding ground. In RNL, the pedestal received shadows from the objects, but the pedestal itself was part of the computer-generated scene. The example we show in this section involves adding several CG objects into the photograph of a museum plaza (seen in Figure 11.38[a]) such that the objects realistically shadow and reflect light with the ground below them (see Figure 11.38[c]). In this example, the light probe image that corresponds to this background plate is the sunny lighting environment shown in Figure 11.14.

Usually, the noticeable effect a new object has on a scene is limited to a local area near the object. We call this area the "local scene," which in the case of the scene in Figure 11.38(a) is the ground in front of the museum. Our strategy for casting shadows on the local scene is to convert the local scene into a CG representation of its geometry and reflectance that can participate in the lighting simulation along with the CG objects. We can then use standard IBL machinery to light the objects and the local scene together by the IBL environment. The local scene model does not need to have the precise geometry and reflectance of what it represents in the real world, but it does need to satisfy two properties. First, it should have approximately the same geometry that the real scene it represents, in that the shadows reveal some of the scene structure. Second, it should have reflectance properties (e.g., texture maps) that cause it to look just like the real local scene when illuminated on its own by the IBL environment.

Modeling the geometry of the local scene can be done by surveying the scene or using photogrammetric modeling techniques (e.g., Debevec et al. [63]) or software (e.g., ImageModeler by Realviz, http://www.realviz.com). Because the geometry needed is often very simple, it can also be modeled by eye in a 3D modeling package. Similar techniques should also be used to recover the camera position, rotation, and focal length used to take the background photograph.

Obtaining the appropriate texture map for the local scene is extremely simple. We first assign the local scene a diffuse white color. We render an image of the white local scene as lit by the IBL lighting environment, as in Figure 11.38(c). We then divide the image of the real local scene (Figure 11.38[a]) in the background plate by the rendering to produce the reflectance image for the local scene (Figure 11.38[d]). Finally, we texture map the local scene with this reflectance image using camera projection (supported in most rendering systems) to project the image onto the local scene geometry. If we create a new IBL rendering of the local scene using this texture map, it will look just as it did in the original background plate. The difference is that it now participates in the lighting calculation and, thus, can receive shadows and interreflect light with synthetic objects added to the scene.

Figure 11.38(e) shows the result of adding a 3D model of a sculpture and four spheres on top of the local scene geometry before computing the IBL rendering. The objects cast shadows on the ground as they reflect and refract the light of the scene. One can also notice bounced beige light near the ground of the beige sphere. The

color and direction of the shadows are consistent with the real shadow cast on the ground by a post to the right of the view of the camera.

The reason the reflectance-solving process works is because the pixel color of a diffuse surface is obtained by multiplying its surface color by the color and intensity of the light arriving at the surface. This calculation was described in detail in Case 3 of Section 11.4. More technically stated, the surface's radiance L is its reflectance ρ times the irradiance E at the surface. Inverting this equation, we have $\rho = L/E$. The background plate image shows the surface radiance L of each pixel of the local scene. The IBL rendering of the diffuse white version of the local scene yields an estimate of the irradiance E at every point on the local scene. Thus, dividing L by the estimate of E yields an image ρ of the surface reflectance of the local scene. By construction, assuming the local scene is convex, these are the reflectance properties that make the local scene look like the background plate when lit by the IBL environment.

This process can be modified slightly for improved results when the local scene is nonconvex or non-Lambertian. For a nonconvex local scene, such as a chair or a staircase, light will reflect between the surfaces of the local scene. That means that our estimate of the irradiance E arriving at the surfaces of the local scene should account for light from the IBL environment as well as interreflected light from the rest of the local scene. The process described earlier does this, but the indirect light is computed as if the local scene were completely white, which will usually overestimate the amount of indirect light received from the other surfaces. As a result, the reflectance of the local scene will be underestimated, and the local scene texture's map will be too dark in concave areas.

FIGURE 11.38 *Adding synthetic objects that cast shadows. (a) A background plate taken near the location of the light probe image in Figure 11.14. It shows the radiance L of pixels in the local scene. (b) A light probe image taken within the scene. (c) A diffuse surface is added to the scene at the position of the ground and is rendered as lit by the surrounding IBL environment. The resulting pixel values on the surface produce an estimate of the irradiance E at each pixel. (d) Dividing (a) by (c) yields the lighting-independent texture map for the local scene. (e) A CG statue and four CG spheres are added on top of the local scene and illuminated by the IBL environment. The CG objects cast shadows and interreflect light with the real scene.*

(a)

(b)

(c)

(d)

(e)

The way to avoid this problem is to assume more accurate reflectance properties for the local scene before computing the irradiance image. Typical surfaces in the world reflect approximately 25% of the incident illumination, and thus assuming

an initial surface reflectance of $\rho_0 = 0.25$ is a more reasonable initial guess. After rendering the local scene, we should compute the irradiance estimate E_0 as $E_0 = L_0/\rho_0$, where L_0 is the rendering generated by lighting the local scene by the IBL environment. This is another simple application of the $L = \rho E$ formula. Then, the texture map values for the local scene can be computed as before, as $\rho_1 = L/E_0$.

If the local scene exhibits particularly significant self-occlusion or spatially varying coloration, the reflectance estimates can be further refined by using the computed reflectance properties ρ_1 as a new initial estimate for the same procedure. We illuminate the new local scene by the IBL environment to obtain L_1, estimate a new map of the irradiance as $E_1 = L_1/\rho_1$, and finally form a new estimate for the per-pixel reflectance ρ_2 of the local scene. Typically, the process converges quickly to the solution after one or two iterations. In fact, this basic process was used to derive the surface reflectance properties of the Parthenon model seen in Figure 11.28, in which the "local scene" was the entire monument [60].

In some cases, we can simplify the surface reflectance estimation process. Often, the local scene is a flat ground plane and the IBL environment surface is distant or infinite. In this case, the irradiance E is the same across the entire surface, and can be computed as the upward-pointing direction of a diffuse convolution of the light probe image. This eliminates the need to render a complete irradiance image such as shown in Figure 11.38(c), and the local scene reflectance is simply its appearance in the background plate divided by the RGB value E.

11.6.1 DIFFERENTIAL RENDERING

The need to use camera projection to map the local scene texture onto its geometry can be avoided using the differential rendering technique described in [58]. In differential rendering, the local scene is assigned (often uniform) diffuse and specular surface reflectance properties similar to the reflectance of the local scene. These properties are chosen by hand or computed using the reflectance estimation process described earlier. Then, two renderings are created: one of the local scene and the objects together (L_{obj}), and one of the local scene without the objects (L_{noobj}). For L_{obj}, an alpha channel image α is created that is 1 for the object pixels and 0 for the nonobject pixels, preferably with antialiasing and transparency encoded as gray

levels. If L_{noobj} and L_{obj} are the same, there is no shadowing. Where L_{obj} is darker, there are shadows, and where L_{obj} is brighter, there are reflections or indirectly bounced light. To apply these photometric effects to the background plate L, we offset its pixel values by the difference between L_{obj} and L_{noobj}. Specifically, the final rendering is computed as

$$L_{\text{final}} = \alpha L_{\text{obj}} + (1 - \alpha)(L + L_{\text{obj}} - L_{\text{noobj}})$$

In this formula, the α mask allows L_{final} to copy the appearance of the objects directly from L_{obj}, and the local scene is rendered using differential rendering.

As an alternative method, we can apply the *ratio* of L_{obj} and L_{noobj} to the background plate, changing the last term of the formula to $L \times L_{\text{obj}}/L_{\text{noobj}}$. If the reflectance properties of the local scene and the IBL environment are modeled accurately, the background plate L and the local scene lit by the IBL environment L_{noobj} would be the same. In this case, the appearance of the local scene in L_{noobj} is copied to L_{final} regardless of whether the difference or the ratio formula is used. When there are inaccuracies in either the lighting or the reflectance, either formula may yield a convincing approximation to the correct result. The difference formula may provide better results for specular reflections and the ratio formula may provide better results for shadows. In either case, only differences between L_{obj} and L_{noobj} will modify the background plate, and where the objects do not affect the local scene, it will look precisely as it did in the background plate.

The benefit of this technique is that the local scene does not need to be projectively texture mapped with its appearance in the background plate image. The drawback is that mirror- and glass-like CG objects will not reflect images of the original local scene. Instead, they will reflect images of the modeled local scene.

11.6.2 RENDERING INTO A NONDIFFUSE LOCAL SCENE

If the local scene is somewhat shiny, we would like the new objects to also appear in reflections in the scene. This was the case for *Fiat Lux* (Figure 11.30), where the marble floor of St. Peter's Basilica is notably specular. The problem is compounded by the fact that a shiny local scene may already have visible specularities from bright

parts of the lighting environment, and these reflections should disappear when virtual objects are placed between the light sources and the observed locations of their specular reflections. Thus, the synthetic local scene needs to model the specular as well as the diffuse reflection characteristics of the real local scene. Unfortunately, estimating spatially varying diffuse and specular reflection components of a surface, even under known illumination, is usually prohibitively challenging for current reflectometry algorithms.

The easiest procedure to follow is to first manually remove visible specular reflections in the local scene using an image-editing program. In *Fiat Lux*, the notable specular reflections in the floor were from the windows, and in some cases, from the lights in the vaulting. Using the edited background plate, we then solve for the local scene reflectance assuming that it is diffuse (as described previously). For the final rendering, we add a specular component to the local scene reflectance whose intensity and roughness are selected by hand to match the appearance of the specularities seen in the local scene on the background plate. The IBL rendering will then show the new CG objects reflecting in the local scene according to the specified specular behavior, and light sources in the IBL environment will also reflect in the local scene when their light is not blocked by the CG objects. This process also provides opportunities for art direction. In *Fiat Lux*, for example, the floor of St. Peter's was chosen to have a more polished specular reflection than it really had to increase the visual interest of its appearance in the animation.

Sometimes, the specular reflectance of a local scene dominates its diffuse reflectance, as would be seen for a steel or black marble floor. In these cases, it can be difficult to remove the specular reflection through image editing. In such a case, the best solution may be to model the reflectance of the local scene by eye, choosing specular intensity and roughness parameters that cause reflections of the IBL environment to match the original appearance of the local scene reasonably well. If the scene is available for photography under controlled lighting, the local scene can be shaded from specular reflections and illuminated from the side to observe its diffuse component. If the reflectance of the local scene is especially complex and spatially varying, such as an ancient stone and metal inlaid mosaic, one could use a technique such as that described in McAllister [214] or Gardner et al. [101] to derive its surface reflectance parameters by analyzing a set of images taken from many incident illumination directions.

11.7 USEFUL IBL APPROXIMATIONS

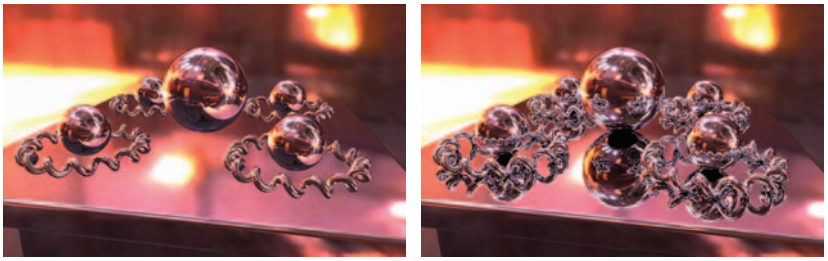
Many of today's rendering programs include specific support for IBL (often referred to as "HDRI"), making IBL a straightforward process to use for many computer graphics applications. However, not every production pipeline is designed to support global illumination, and real-time applications require faster rendering times than ray-traced illumination solutions typically allow. Fortunately, there are several approximate IBL techniques that allow particularly fast rendering times and that can be implemented within more traditional rendering pipelines. The sections that follow describe two of them: *environment mapping* and *ambient occlusion*. The discussion includes the advantages and disadvantages of these two approaches.

11.7.1 ENVIRONMENT MAPPING

Environment mapping [23,222,361,113] is a forerunner of IBL in which an omnidirectional LDR image of an environment is directly texture mapped onto an object surface to produce the appearance of it reflecting the environment. The omnidirectional *environment map* or *reflection map* image can also be *preconvolved* by a blurring filter to simulate the reflections from rough specular or diffuse surfaces. The environment map is mapped onto the object according to the surface normal of each point, which makes the rendering process extremely fast once the appropriate reflection map images have been computed. The disadvantage is that the technique does not take into account how light is shadowed and interreflects between object surfaces, which can be an impediment to realism. For objects that are relatively shiny and convex, the errors introduced by the approximation can be insignificant. For objects with more complex geometry and reflectance properties, however, the results can be less realistic.

Environment mapping is most successful and most often used for simulating the specular reflections of an object. In this case, for each surface point, the ray from the camera R is reflected about the surface normal N to determine the reflected vector R', computed as follows.

$$R' = R - 2(R \cdot N)N$$



(a) (b)

FIGURE 11.39 *(a) A shiny version of the test scene rendered using environment mapping. (b) A shiny version of the scene rendered with ray-tracing-based IBL, producing interreflections.*

Then, the point on the object is drawn with the pixel color of the environment image corresponding to the direction of R'. Figure 11.39(a) shows this environment mapping process applied to the scene shown in Figure 11.1.

The environment-mapped rendering gives the appearance that the objects are reflecting the environment. However, the appearance is somewhat strange because we do not see reflections of the sphere in the table (the spheres appear to float above it). We can compare this rendering to the corresponding IBL rendering of mirror-like objects in Figure 11.39(b), which exhibits appropriate interreflections. If the scene were a single convex object, however, the two renderings would be the same.

It is interesting to note that this form of environment mapping does not require that the environment map be higher in its dynamic range than the final display. Because every pixel in the rendering comes directly from the environment map, clipping the pixel values of the environment map image would be unnoticeable on a similarly clipped display, unless the rendering were to exhibit significant motion blur or image defocus.

Environment mapping can also be used to simulate the reflection of an environment by surfaces with nonmirror reflectance properties, by *preconvolving* the image of the environment by various convolution filters [222,113,30,124]. This takes advantage of an effect noted by Ramamoorthi and Hanrahan [262]: that a

detailed environment reflected in a rough specular surface looks similar to a blurred environment reflected in a mirror-like surface. Often, a specular Phong cosine lobe [254] is used as the convolution filter.

To simulate Lambertian diffuse reflection with environment mapping, a hemispherical cosine lobe is used as the convolution filter, yielding an *irradiance environment map*. For diffuse reflection, one indexes into the irradiance environment map using the object point's surface normal direction N rather than the reflected vector R'. Convolving the image can be computationally expensive, but because irradiance images lack sharp detail, a close approximation can be made by convolving a low-resolution version of as few as 32×16 pixels in latitude–longitude format. Cabral et al. [30] suggested that such a convolution could be performed efficiently on a spherical harmonic (SH) decomposition of the incident illumination, and Ramamoorthi and Hanrahan [262] noted that computing the SH reconstruction of an environment using the first nine terms (orders 0, 1, and 2) of the SH decomposition approximates the diffuse convolution of any lighting environment to within 99% accuracy.[8]

Figure 11.41(a) shows a scene rendered using diffuse environment mapping. Because environment mapping does not simulate self-shadowing, the image is not as realistic a rendering of the scene as the IBL solution shown in Figure 11.41(d). However, if the scene were a single convex object, the two renderings would again be the same. In general, environment mapping produces more convincing results for specular reflection than for diffuse reflection. As a result, for objects with both specular and diffuse reflectance components, it is common for environment mapping to be used for the specular reflection and traditional lighting for the diffuse component. The technique of *reflection occlusion* [173] can further increase the realism of specular environment mapping by tracing reflected rays from each surface point to determine if the reflection of the environment should be omitted because of self-occlusion.

When the incident illumination is blurred by a convolution filter, it becomes necessary that the environment map covers the full dynamic range of the incident illumination to obtain accurate results. Figure 11.40 shows a comparison of using

8 This technique for computing an irradiance environment map can yield regions with negative pixel values when applied to an environment with concentrated light sources because of the Gibbs ringing phenomenon.

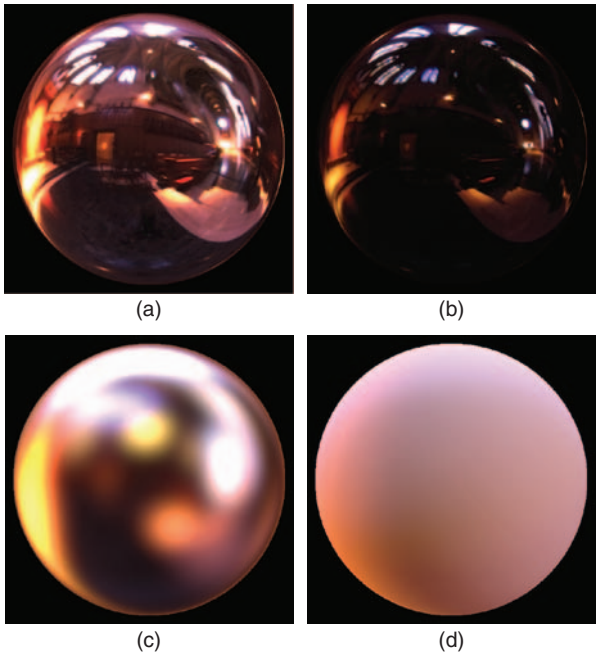


FIGURE 11.40 (a) HDR Grace Cathedral light probe image in the mirrored sphere format. (b) The light probe in (a) shown with lower exposure, revealing details in the bright regions. (c) A specular convolution of (a). (d) A diffuse convolution of (a), showing how this lighting environment would illuminate a diffuse sphere. (e) An LDR environment map version of (a) with clipped pixel values. (f) The image in (e) with lower exposure, showing that the highlights have been clipped. (g) A specular convolution of (e), showing inaccurately reduced highlight size and intensity relative to (c). (h) A diffuse convolution of (e), yielding an inaccurately dark and desaturated rendering compared with (d).

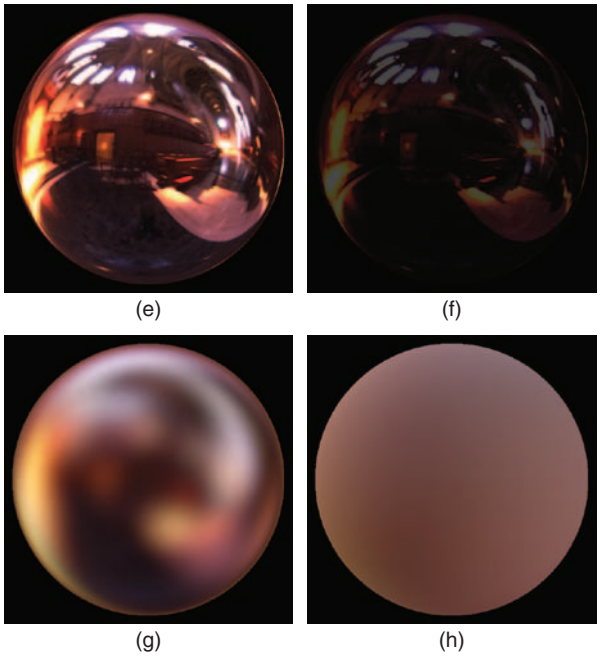


FIGURE 11.40 (*continued*)

an LDR environment map versus an HDR light probe image for rendering a diffuse sphere using convolution and environment mapping.

11.7.2 AMBIENT OCCLUSION

Ambient occlusion [173] can be used to approximate IBL using a single-bounce irradiance calculation under the assumption that the IBL lighting environment is relatively even. The technique leverages the key IBL step of firing rays out from

object surfaces to estimate the amount of light arriving from the visible parts of the environment, but it uses diffuse environment mapping to determine the coloration of the light at the surface point. The result is an approximate but efficient IBL process that can perform well with artistic guidance and that avoids noise from the light sampling process.

The first step in ambient occlusion is to use an IBL-like process to render a neutral diffuse version of the scene as illuminated by a homogeneously white illumination environment. In this step, the surfaces of the scene are set to a neutral diffuse reflectance so that the rendered image produces pixel values that are proportional to a surface point's irradiance. This step can be performed using the Monte Carlo ray-tracing process (described in Section 11.4) or by converting the white lighting environment into a constellation of light sources (Section 11.5.2). In the latter case, the rendering can be formed by adding together scan-line renderings of the scene lit from each lighting direction, with the shadows being calculated by a shadow buffer algorithm. Usually, no additional light bounces are simulated when computing this rendering. This estimate of the irradiance from the environment at each point is called the "ambient occlusion map," seen in Figure 11.41(b).

The next step is to multiply the ambient occlusion map by an image of the scene environment mapped with the diffuse convolution of the lighting environment, as in Figure 11.41(a). The product, seen in Figure 11.41(c), applies the self-shadowing characteristics of the ambient occlusion map to the diffuse lighting characteristics of the environment-mapped scene, yielding a rendering that is considerably more realistic. If the scene has different surface colors, this rendering can be multiplied by an image of the diffuse color of each point in the scene, which approximates having differently colored surfaces during the original rendering. If needed, specular reflections can be added using either ray tracing or specular environment mapping.

Ambient occlusion does not precisely reproduce how a scene would appear as illuminated by the light probe using standard IBL. We can see, for example, differences in comparing Figures 11.41(c) and 11.41(d). Most notably, the shadows in the ambient occlusion rendering are much softer than they appear in the standard IBL rendering. The reason is that the ambient occlusion rendering is computed as if from a completely diffuse lighting environment, whereas a standard IBL rendering computes which specific parts of the lighting environment become occluded for each part of the shadowed area. The ambient occlusion result can be improved

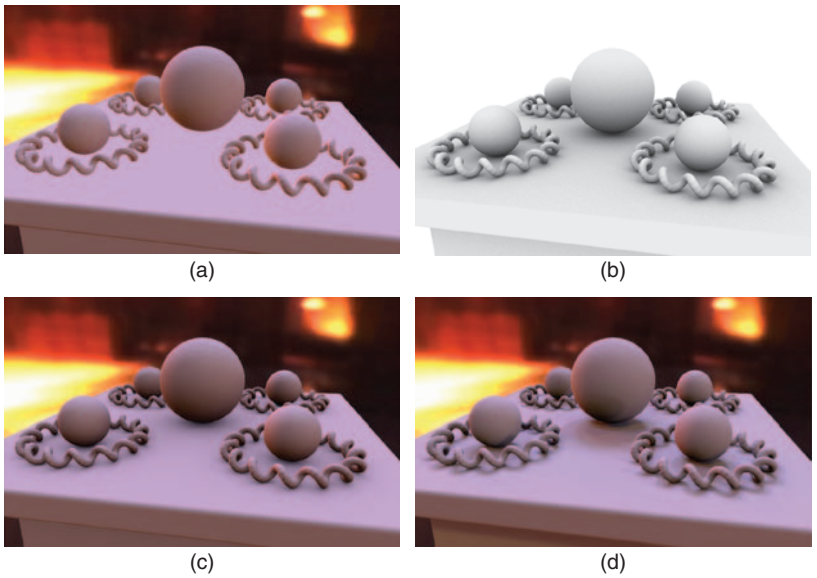


FIGURE 11.41 (a) *A scene environment mapped with the diffuse convolution of the Grace Cathedral environment.* (b) *An ambient occlusion map obtained using IBL to light the scene with a homogeneous white lighting environment.* (c) *The ambient occlusion map multiplied by* (a). (d) *The scene illuminated using standard IBL from the light probe image. For this scene, the principal difference from* (c) *is in the shadow regions, which have more directional detail in* (d).

to some extent using *bent normals* [173], where the diffuse convolution of the light probe image is mapped onto the object surfaces according to the average direction of unoccluded light, rather than the true surface normal. However, because the surface colors are still sampled from a diffuse convolution of the light probe image, the ambient occlusion rendering will lack the shading detail obtainable from sampling the light probe image directly.

Ambient occlusion most accurately approximates the correct lighting solution when the lighting environment is relatively diffuse. In this case, the homogeneous environment used to compute the occlusion is a close approximation to the environment desired to light the scene. Ambient occlusion is not designed to simulate light from environments that include concentrated light sources, as the directional detail of the environment is lost in the diffuse convolution process. For IBL environments that do have concentrated light sources, an effective way of handling them is to simulate them as direct light sources (as described in Section 11.5.1), delete them from the IBL environment, and use a diffuse convolution of the modified IBL environment to multiply the ambient occlusion map.

Although computing ambient occlusion maps requires sending out a multitude of rays to the lighting environment, the number of rays that need to be sent is minimized because the environment has minimal variance, which alleviates the sampling problem. Also, the ambient occlusion map is solely a function of the object geometry and is independent of the lighting environment. Because of this, the technique can be used to render an object with different lighting environments while performing the ambient occlusion calculation map only once. This makes real-time implementations very fast, especially for rotating lighting environments for which performing additional diffuse convolutions is also unnecessary. In addition, the technique allows for relighting effects to be performed inside a standard compositing system. For example, the convolved light probe image can be manually edited and a relit version of the scene can be created quickly using the preexisting normals and ambient occlusion map without rerendering.

11.8 IMAGE-BASED LIGHTING FOR REAL OBJECTS AND PEOPLE

The IBL techniques described so far are useful for lighting synthetic objects and scenes. It is easy to imagine uses for a process that could illuminate *real* scenes, objects, and people with IBL environments. To do this, one could attempt to build a virtual model of the desired subject's geometry and reflectance, and then illuminate the model using the IBL techniques already presented. However, creating photoreal models of the geometry and reflectance of objects (and particularly people) is a difficult process, and a more direct route would be desirable. In fact, there is a

straightforward process for lighting real subjects with IBL that requires only a set of images of the subject under a variety of directional lighting conditions.

11.8.1 A TECHNIQUE FOR LIGHTING REAL SUBJECTS

The technique is based on the fact that light is *additive*, which can be described simply as follows. Suppose we have two images of a subject, one lit from the left and one lit from the right. We can create an image of the subject lit with both lights at once simply by adding the two images together, as demonstrated by Moon and Spencer [118]. If the image pixel values are proportional to the light in the scene, this process yields exactly the right answer, with all of the correct shading, highlights, and shadows the scene would exhibit under both light sources. Furthermore, the color channels of the two images, for example, can be independently scaled before they are added, allowing one to virtually light the subject with a bright orange light to the right and a dim blue light to the left.

As we have seen in Section 11.5.2, an IBL lighting environment can be simulated as a constellation of light sources surrounding the subject. If one could quickly light a person from a dense sampling of directions distributed across the entire sphere of incident illumination, it should be possible to recombine tinted and scaled versions of these images to show how the person would look in any lighting environment. The light stage device described by Marcelja [59] (Figure 11.42) is designed to acquire precisely such a data set. The 250-W halogen spotlight of the device is mounted on a two-axis rotation mechanism such that the light can spiral from the top of the sphere to the bottom in approximately 1 min. During this time, a set of digital video cameras can record the subject's appearance, as illuminated by hundreds of lighting directions distributed throughout the sphere. A subsampled light stage data set of a person's face is seen in Figure 11.43(a).

Figure 11.43(c) shows the Grace Cathedral lighting environment remapped to be the same resolution and in the same longitude–latitude space as the light stage data set. For each image of the face in the data set, the remapped environment indicates the color and intensity of the light from the environment in the corresponding direction. Thus, we can multiply the red, green, and blue color channels of each light stage image by the amount of red, green, and blue light in the corresponding direction in the lighting environment to obtain a modulated image data

FIGURE 11.42 *(a) The Light Stage 1 device for lighting a person's face from the full sphere of incident illumination directions. (b) A 1-min exposure taken during a data set acquisition. Images recorded from the right video camera are shown in Figure 11.43.*

set, as in Figure 11.43(d). Adding all of these images together then produces an image of the subject as illuminated by the complete lighting environment, as seen in Figure 11.44(a). Results obtained for three more lighting environments are shown in Figures 11.44(b) through 11.44(d).

11.8.2 RELIGHTING FROM COMPRESSED IMAGE DATA SETS

Computing the weighted sum of the light stage images is a simple computation, but it requires accessing a large amount of data to create each rendering. This process can be accelerated by performing the computation on compressed versions of the original images. In particular, if the images are compressed using

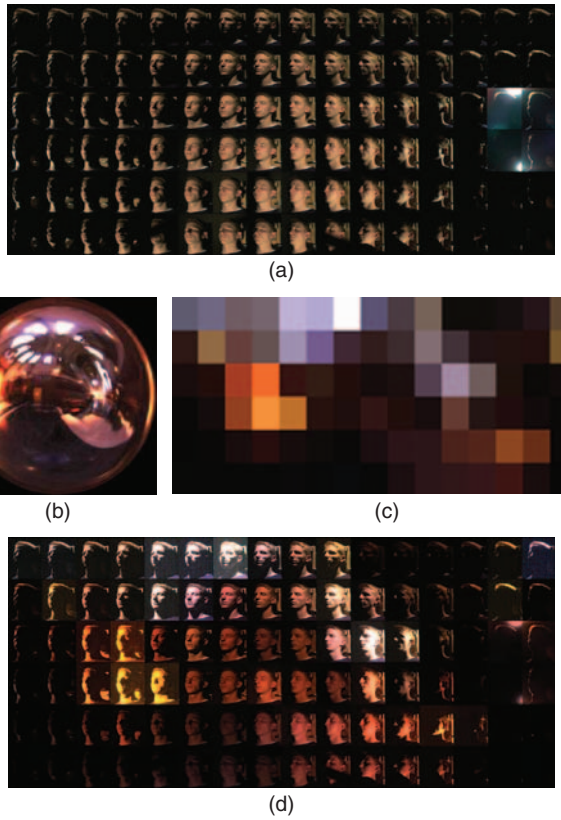


FIGURE 11.43 (a) Light stage data of a face illuminated from the full sphere of lighting directions. The image shows 96 images sampled from the 2000-image data set. (b) The Grace Cathedral light probe image. (c) The Grace probe resampled into the same longitude–latitude space as the light stage data set. (d) Face images scaled according to the color and intensity of the corresponding directions of illumination in the Grace light probe. Figure 11.44 (a) shows the face illuminated by the Grace probe created by summing these scaled images.

FIGURE 11.44 *Renderings of the light stage data set from Figure 11.43 as illuminated by four IBL environments: (a) Grace cathedral, (b) eucalyptus grove, (c) Uffizi Gallery, and (d) St. Peter's Basilica.*

an orthonormal transform such as the discrete cosine transform (DCT), the linear combination of the images can be computed directly on the basis coefficients of the compressed images [301]. The downloadable Facial Reflectance Field Demo [318] (http://www.debevec.org/FaceDemo/) uses DCT-compressed versions of light stage data sets to allow a user to relight a face interactively using either light probe images or user-controlled light sources in real time.

A light stage data set can be parameterized by the four dimensions of image coordinates (u, v) and lighting directions (θ, ϕ). Choosing a particular pixel (u, v) on the subject, we can create a small image (called the "reflectance function" of the pixel) from the color the pixel reflects toward the camera for all incident lighting directions (θ, ϕ) (Figure 11.45). In the Facial Reflectance Field Demo, the 4D light stage data sets are actually DCT compressed in the lighting dimensions rather than the spatial dimensions, exploiting coherence in the reflectance functions rather than in the images themselves. When the DCT coefficients of the reflectance functions

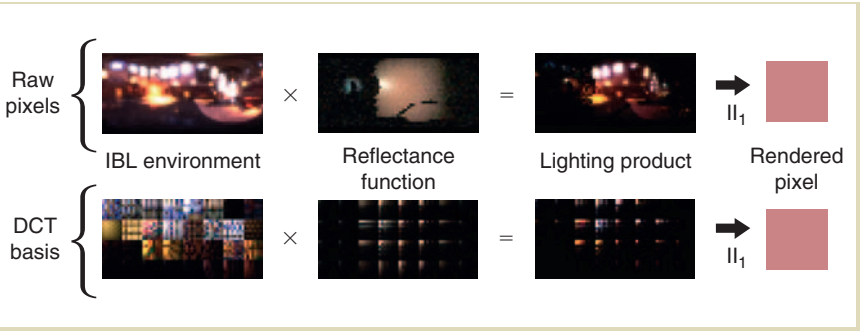


FIGURE 11.45 *Relighting a reflectance function can be performed on the original pixel values (top row) or on DCT coefficients of the illumination data.*

are quantized (as in JPEG compression), up to 90% of the data maps to zero and can be skipped in the relighting calculations, enabling real-time rendering. The process of relighting a single pixel of a light stage data set based on its reflectance function is shown in Figure 11.45.

This image-based relighting process can also be applied in the domain of computer-generated objects. One simply needs to render the object under an array of different lighting conditions to produce a virtual light stage data set of the object. This can be useful in that the basis images can be rendered using high-quality off-line lighting simulations but then recombined in real time through the relighting process, maintaining the quality of the off-line renderings. In the context of CG objects, the content of a reflectance function for a surface point is its *precomputed radiance transfer*. Sloan et al. [295], Ramamoorthi and Hanrahan [262], and Ng et al. [240] have noted that the basis lighting conditions need not be rendered with point source illumination. Specifically, Sloan et al. [295] and Ramamoorthi and Hanrahan [262] used the SH basis, whereas Ng et al. [240] used a wavelet basis. These techniques demonstrate varying the viewpoint of the object by mapping its radiance transfer characteristics onto a 3D geometric model of the object. Whereas these earlier techniques have been optimized for diffuse surface reflectance in low-frequency lighting environments, Liu et al. [192] use both a wavelet representation and

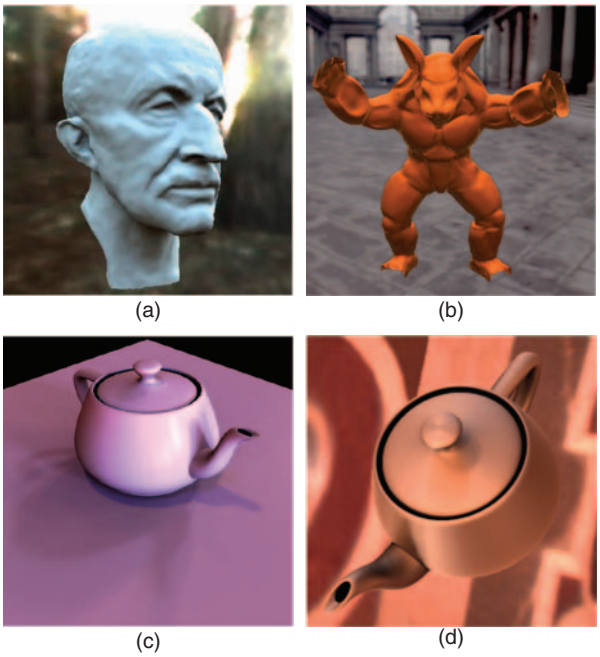


FIGURE 11.46 Interactive IBL renderings of 3D objects using basis decompositions of the lighting environment and surface reflectance functions. (a) Max Planck model rendered in the RNL environment using precomputed radiance transfer [295] based on SHs. (b) Armadillo rendered in the Uffizi Gallery using a SH reflection map [262]. (c) Teapot rendered into Grace Cathedral using a 2D Haar transform [240] of the lighting and reflectance, achieving detailed shadows. (d) Teapot rendered into St. Peter's Basilica using precomputed radiance transfer represented by Haar wavelets and compressed with clustered principal component analysis [192] to produce sharp shadow detail, as seen on the teapot lid.

clustered principal components analysis of precomputed radiance transfer (PRT) functions to render view-dependent reflections from glossy objects in high-frequency lighting environments, producing sharp shadows from light sources interactively. Sample renderings made using these techniques are shown in Figure 11.46. Using the GPU to perform image-based relighting on CG objects with techniques such as these promises to become the standard method for using IBL in video games and other interactive rendering applications.

11.9 CONCLUSIONS

In this chapter, we have seen how HDR images can be used as sources of illumination for both computer-generated and real objects and scenes through IBL. Acquiring real-world lighting for IBL involves taking omnidirectional HDR images through one of several techniques, yielding a data set of the color and intensity of light arriving from every direction in the environment. This image of the incident illumination is mapped onto a surface surrounding the object, and a lighting simulation algorithm is used to compute how the object would appear as if lit by the captured illumination. With the appropriate optimizations, such images can be computed efficiently using either global illumination or traditional rendering techniques, and recent techniques have allowed IBL to happen in real-time. Real objects can be illuminated by new environments by capturing how they appear under many individual lighting conditions and then recombining them according to the light in an IBL environment.

The key benefit of IBL is that it provides a missing link between light in the real world and light in the virtual world. With IBL, a ray of light can be captured by an HDR camera, reflected from a virtual surface in a rendering algorithm, and turned back into real light by a display. The IBL process has a natural application wherever it is necessary to merge CG imagery into real scenes, in that the CG can be lit and rendered as if it were actually there. Conversely, the light stage technique allows IBL to illuminate real-world objects with the light of either virtual or real environments. In its applications so far, IBL has rendered virtual creatures into real movie locations, virtual cars onto real roads, virtual buildings under real skies, and real actors into virtually created sets. For movies, video games, architecture, and design, IBL can connect what is real with what can only be imagined.

This page intentionally left blank

Appendix A

LIST OF SYMBOLS

Symbol	Section	Description
$\otimes$	10.7.3	Convolution operator
α	2.7	A channel of $L\alpha\beta$ color space
α	7.6.1	The key of a scene
β	2.7	A channel of $L\alpha\beta$ color space
γ	2.10	Exponent used for gamma correction
σ	7.2	Semisaturation constant
a	2.9	Color opponent channel of $L^*a^*b^*$ color space, also color opponent channel used in CIECAM02
A	2.5	CIE standard illuminant approximating incandescent light
A	2.9	Achromatic response, computed in CIECAM02
b	2.9	Color opponent channel of $L^*a^*b^*$ color space, also color opponent channel used in CIECAM02
B	2.5	CIE standard illuminant approximating direct sunlight
c	2.9	Viewing condition parameter used in CIECAM02
C	2.5	CIE standard illuminant approximating indirect sunlight
C	2.9	Chroma

(Continued)

© 2010, Elsevier Inc. All rights reserved.
DOI: 10.1016/B978-0-12-374914-7.00012-9

Symbol	Section	Description
C^*_{ab}	2.9	Chroma, computed in $L^*a^*b^*$ color space
C^*_{uv}	2.9	Chroma, computed in $L^*u^*v^*$ color space
L_v		
D	2.9	Degree of adaptation, used in CIECAM02
D_{55}	2.5	CIE standard illuminant with a correlated color temperature of 5503 Kelvin (K)
D_{65}	2.5	CIE standard illuminant with a correlated color temperature of 6504 Kelvin (K)
D_{75}	2.5	CIE standard illuminant with a correlated color temperature of 7504 Kelvin (K)
ΔE^* 1994	3.4	CIE color difference metric
ΔE^*_{ab}	2.9	Color difference measured in $L^*a^*b^*$ color space
ΔE^*_{uv}	2.9	Color difference measured in $L^*u^*v^*$ color space
e	2.9	Eccentricity factor, used in CIECAM02
E	2.5	CIE equal-energy illuminant
E_e	2.1	Irradiance, measured in Watts per square meter
E_v	2.2	Illuminance, measured in lumen per square meter
F	2.9	Viewing condition parameter used in CIECAM02
F_2	2.5	CIE standard illuminant approximating fluorescent light
F_L	2.9	Factor modeling partial adaptation, computed using the adapting field luminance in CIECAM02
h	2.9	Hue angle as used in CIECAM02
h_{ab}	2.9	Hue, computed in $L^*a^*b^*$ color space
h_{uv}	2.9	Hue, computed in $L^*u^*v^*$ color space
H	2.9	Appearance correlate for hue
I	2.11, 7	Catch-all symbol used to indicate an arbitrary value

(Continued)

Symbol	Section	Description
I_e	2.1	Radiant intensity, measured in Watts per steradian
I_v	2.2	Luminous intensity, measured in lumen per steradian or candela
J	2.9	Appearance correlate for lightness
L	2.2	Luminance
L_a	2.9	Adapting field luminance
L_d	7.6.1	Display luminance
L_e	2.1	Radiance, measured in Watts per steradian per square meter
L_v	2.2	Luminance, measured in candela per square meter
L_w	7.6.1	World or scene luminance (also Y_W)
$L\alpha\beta$	2.7	Color opponent space
LMS	2.5	Color space approximating the output of cone photoreceptors
$L^*a^*b^*$	2.9	CIE color space, also known as CIELAB
$L^*u^*v^*$	2.9	CIE color space, also known as CIELUV
M	2.9	Appearance correlate for colorfulness
M_{Bradford}	2.5	Bradford chromatic adaptation transform
M_{CAT02}	2.5	CAT02 chromatic adaptation transform
M_e	2.1	Radiant exitance, measured in Watts per square meter
M_H	2.9	Hunt–Pointer–Estevez transformation matrix
$M_{\text{von Kries}}$	2.5	von Kries chromatic adaptation transform
M_v	2.2	Luminous exitance, measure in lumen per square meter
N_c	2.9	Viewing condition parameter used in CIECAM02
P_e	2.1	Radiant power, measured in Watts (W) or Joules per second
P_v	2.2	Luminous power, measured in lumen (lm)

(Continued)

Symbol	Section	Description
Q	2.9	Appearance correlate for brightness
Q_e	2.1	Radiant energy, measured in Joules (J)
Q_v	2.2	Luminous energy, measured in lumen per second
R	7.2	Photoreceptor response
RGB	2.4	A generic red, green, and blue color space
$R_d G_d B_d$	7.6.2	Red, green, and blue values scaled within the displayable range
$R_W G_W B_W$	2.4	Red, green and blue values referring to a world or scene color
s	7.6.2	Saturation parameter
s	2.9	Appearance correlate for saturation
s_{uv}	2.9	Saturation, computed in $L^*u^*v^*$ color space
t	2.9	Magnitude factor, used in CIECAM02
T	2.5	Correlated color temperature, measured in Kelvin (K)
$V(\lambda)$	2.2	CIE photopic luminous efficiency curve
XYZ	2.3	CIE-defined standard tristimulus values
xyz	2.3	Normalized XYZ tristimulus values
Y	2.4	Y component of an XYZ tristimulus value, indicating CIE luminance
Y_W	3.3	World or scene luminance (also L_w)
Y_b	2.9	Relative background luminance
YC_BC_R	2.7	Color opponent space used for the JPEG file format

Bibliography

[1] DICOM PS 3-2004. "Part 14: Grayscale Standard Display Function," in *Digital Imaging and Communications in Medicine (DICOM)*, Rosslyn, VA, National Electrical Manufacturers Association, 2004, *http://medical.nema.org/dicom/2004/04_14PU.PDF*.

[2] E. H. Adelson. "Saturation and Adaptation in the Rod System," *Vision Research*, 22:1299–1312, 1982.

[3] Adobe. Tiff 6.0 specification, 1992, *http://partners.adobe.com/asn/tech/tiff/specification.jsp*.

[4] Adobe. Digital negative (DNG), 2004, *http://www.adobe.com/products/dng/main.html*.

[5] S. Agarwal, R. Ramamoorthi, S. Belongie, and H. W. Jensen. "Structured Importance Sampling of Environment Maps," *ACM Transactions on Graphics*, 22(3):605–612, July 2003.

[6] A. O. Akyüz, R. Fleming, B. Riecke, E. Reinhard, and H. Bülthoff. "Do HDR Displays Support LDR Content? A Psychophysical Evaluation," *ACM Transactions on Graphics*, 26(3):38, 2007.

[7] A. O. Akyüz, E. A. Khan, and E. Reinhard. "Ghost Removal in High Dynamic Range Images," in *IEEE International Conference on Image Processing*, pp. 2005–2008, October 2006.

© 2010, Elsevier Inc. All rights reserved.
DOI: 10.1016/B978-0-12-374914-7.00013-0

[8] A. O. Akyüz and E. Reinhard. "Color Appearance in High Dynamic Range Imaging," *SPIE Journal of Electronic Imaging*, 15(3):033001-1–033001-12, 2006.

[9] A. O. Akyüz and E. Reinhard. "Noise Reduction in High Dynamic Range Maging," *The Journal of Visual Communication and Image Representation*, 18(5):366–376, 2007.

[10] A. O. Akyüz, E. Reinhard, and S. Pattanaik. "Color Appearance Models and Dynamic Range Reduction," in *First ACM Symposium on Applied Perception in Graphics and Visualization (APGV)*, pp. 166, 2004.

[11] A. O. Akyüz, E. Reinhard, E. A. Khan, and G. M. Johnson. *Color Imaging: Fundamentals and Applications*. Wellesley, MA: AK Peters, 2008.

[12] M. Ashikhmin. "A Tone Mapping Algorithm for High-Contrast Images," in *Proceedings of 13th Eurographics Workshop on Rendering*, pp. 145–155, 2002.

[13] M. Ashikhmin and P. Shirley. "An Anisotropic Phong BRDF Model," *Journal of Graphics Tools*, 5(2):25–32, 2000.

[14] V. Aurich and J. Weule. "Non-linear Gaussian Filters Performing Edge Preserving Diffusion," in *Proceedings of the DAGM Symposium*, 1995.

[15] T. O. Aydın, R. Mantiuk, K. Myszkowski, and H.-P. Seidel. "Dynamic Range Independent Image Quality Assessment," *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 27(3), 2008. Article 69.

[16] T. O. Aydın, R. Mantiuk, and H.-P. Seidel. "Extending Quality Metrics to Full Luminance Range Images," in A. Heimer (ed.), *Human Vision and Electronic Imaging XIII*, pp. 68060B1–10, San Jose, CA: SPIE, 2008.

[17] T. O. Aydın, K. Myszkowski, and H.-P. Seidel. "Predicting Display Visibility under Dynamically Changing Lighting Conditions," *Computer Graphics Forum (Proceedings of EUROGRAPHICS)*, 28(3):173–182, 2009.

[18] F. Banterle, P. Ledda, K. Debattista, and A. Chalmers. "Inverse Tone Mapping," in J. Kautz and S. Pattanaik (eds.), *18th Eurographics Symposium on Rendering*, pp. 321–326, Natick, MA: AK Peters, 2007.

[19] K. Barnard. "Practical Colour Constancy," Ph.D. thesis, Simon Fraser University, School of Computing, 1999.

[20] P. G. J. Barten. *Contrast Sensitivity of the Human Eye and its Effects on Image Quality*. SPIE – The International Society for Optical Engineering, Bellingham, WA, United States, 1999.

[21] O. Bimber and D. Iwai. "Superimposing Dynamic Range," *ACM Transactions on Graphics*, 27(5):1–8, 2008.

[22] O. M. Blackwell and H. R. Blackwell. "Visual Performance Data for 156 Normal Observers of Various Ages," *Journal of the Illuminating Engineering Society*, 1(1):3–13, 1971.

[23] J. F. Blinn. "Texture and Reflection in Computer-Generated Images," *Communications of the ACM*, 19(10):542–547, October 1976.

[24] M. R. Bolin and G. W. Meyer. "A Perceptually Based Adaptive Sampling Algorithm," in *Proceedings of SIGGRAPH*, pp. 299–310, 1998.

[25] A. C. Bovik, J. D. Gibson, and Al Bovik, eds. *Handbook of Image and Video Processing*. Orlando, FL, Academic Press, 2000.

[26] A. P. Bradley. "A Wavelet Visible Difference Predictor," *IEEE Transactions on Image Processing*, 8(5):717–730, 1999.

[27] D. Burr. "Implications of the Craik-O'Brien Illusion for Brightness Perception," *Vision Research*, 27(11):1903–1913, 1987.

[28] P. J. Burt and E. H. Adelson. "The Laplacian Pyramid as a Compact Image Code," *IEEE Transactions on Communications*, COM-31(4):532–540, 1983.

[29] P. J. Burt and E. H. Adelson. "A Multiresolution Spline with Application to Image Mosaics," *ACM Transactions on Graphics*, 2(4):217–236, 1983.

[30] B. Cabral, N. Max, and R. Springmeyer. "Bidirectional Reflection Functions from Surface Bump Maps," in *Computer Graphics (Proceedings of SIGGRAPH 87)*, Vol. 21, pp. 273–281, July 1987.

[31] T. Caelli, M. Hubner, and I. Rentschler. "Detection of Phase-Shifts in Two-Dimensional Images," *Perception and Psychophysics*, 37:536–542, 1985.

[32] A. J. Calabria and M. D. Fairchild. "Perceived Image Contrast and Observer Preference I: The Effects of Lightness, Chroma, and Sharpness Manipulations on Contrast Perception," *Journal of Imaging Science and Technology*, 47:479–493, 2003.

[33] M. W. Cannon, Jr. "Perceived Contrast in the Fovea and Periphery," *Journal of the Optical Society of America A*, 2(10):1760–1768, 1985.

[34] C. R. Carlson, E. H. Adelson, and C. H. Anderson. System for coring an image-representing signal. US Patent 4,523,230. United States Patent and Trademark Office, 1985.

[35] E. Caviedes and F. Oberti. "Chapter 10: No-reference Quality Metric for Degraded and Enhanced Video," in H. R. Wu and K. R. Rao (eds.), *Digital Video Image Quality and Perceptual Coding*, pp. 305–324, 2006.

[36] E. Chen. "QuickTime VR: An Image-based Approach to Virtual Environment Navigation," in *SIGGRAPH 95: Proceedings of the 2nd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 29–38, New York, NY, USA: ACM, 1995.

[37] H. Chen, J. Sung, T. Ha, Y. Park, and C. Hong. "Backlight Local Dimming Algorithm for High Contrast LCD-TV," in *Proceedings Asian Symposium on Information Display*, 2006.

[38] J. Chen, S. Paris, and F. Durand. "Real-Time Edge-Aware Image Processing with the Bilateral Grid," *ACM Transactions on Graphics*, 26(3):103, 2007.

[39] K. Chiu, M. Herf, P. Shirley, S. Swamy, C. Wang, and K. Zimmerman. "Spatially Nonuniform Scaling Functions for High Contrast Images," in *Proceedings of Graphics Interface*, pp. 245–253, May 1993.

[40] P. Choudhury and J. Tumblin. "The Trilateral Filter for High Contrast Images and Meshes," in P. Dutré, F. Suykens, P. Christensen and D. Cohen-Or (eds.) *Proceedings of the Eurographics Symposium on Rendering*, Eurographics Association, pp. 186–196, 2003.

[41] CIE. "An Analytic Model for Describing the Influence of Lighting Parameters upon Visual Performance: Vol. 1, Technical Foundations," Technical report, CIE Pub. 19/2.1 Technical committee 3.1, 1981.

[42] CIE. "The CIE 1997 Interim Colour Appearance Model (Simple Version)," CIECAM97s. Technical report, CIE Pub. 131, Vienna, 1998.

[43] CIE. 15:2004, Colorimetry, Section 9.5. Technical report, CIE, Vienna, 2004.

[44] J. M. Cohen. "Estimating Reflected Radiance Under Complex Distant Illumination," Technical Report RH-TR-2003-1, Rhythm and Hues Studios, 2003.

[45] J. M. Cohen and P. Debevec. "The LightGen HDRShop Plug-in," 2001, *www.hdrshop.com/main-pages/plugins.html*.

[46] P. Corriveau. "Chapter 4: Video Quality Testing," in H. R. Wu and K. R. Rao (eds.), *Digital Video Image Quality and Perceptual Coding*, pp. 125–153, 2006.

[47] M. Cowan, G. Kennel, T. Maier, and B. Walker. "Contrast Sensitivity Experiment to Determine the Bit Depth for Digital Cinema," *SMPTE Motion Imaging Journal*, 113:281–292, 2004.

[48] F. C. Crow. "Summed-area Tables for Texture Mapping," in *Computer Graphics (Proceedings of SIGGRAPH 84)*, Vol. 18, pp. 207–212, July 1984.

[49] S. Daly. "The Visible Difference Predictor: An Algorithm for the Assessment of Image Fidelity," in A. B. Watson (ed.), *Digital Images and Human Vision*, pp. 179–206, Cambridge, MA: MIT Press, 1993.

[50] S. Daly and X. Feng. "Bit-Depth Extension using Spatiotemporal Microdither Based on Models of the Equivalent Input Noise of the Visual System," in *Color Imaging VIII: Processing, Hardcopy, and Applications*, Vol. 5008, pp. 455–466, SPIE, 2003.

[51] S. Daly and X. Feng. "Decontouring: Prevention and Removal of False Contour Artifacts," in *Proceedings of Human Vision and Electronic Imaging IX*, Vol. 5292, SPIE, pp. 130–149, 2004.

[52] G. Damberg, H. Seetzen, G. Ward, M. Kang, P. Longhurst, W. Heidrich, and L. Whitehead. "High-Dynamic-Range Projector," Siggraph Emerging Technologies, August 2007.

[53] H. J. A. Dartnall, J. K. Bowmaker, and J. D. Mollon. "Human Visual Pigments: Microspectrophotometric Results from the Eyes of Seven Persons," *Proceedings of the Royal Society of London B*, 220:115–130, 1983.

[54] H. Davson. *Physiology of the Eye,* 5th ed., Oxford, UK: Pergamon Press, 1990.

[55] R. L. De Valois and K. K. De Valois. "On a Three-Stage Color Model," *Vision Research*, 36:833–836, 1996.

[56] R. L. De Valois and K. K. De Valois. *Spatial Vision*. Oxford: Oxford University Press, 1990.

[57] P. Debevec. "Light Probe Image Gallery," 1999, *http://www.debevec.org/Probes/*.

[58] P. Debevec. "Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-Based Graphics with Global Illumination and High Dynamic Range Photography," in *Proceedings of SIGGRAPH 98*, Computer Graphics Proceedings, Annual Conference Series, pp. 189–198, July 1998.

[59] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. "Acquiring the Reflectance Field of a Human Face," *Proceedings of SIGGRAPH 2000*, pp. 145–156, July 2000.

[60] P. Debevec, C. Tchou, A. Gardner, T. Hawkins, A. Wenger, J. Stumpfel, A. Jones, C. Poullis, N. Yun, P. Einarsson, T. Lundgren, P. Martinez, and M. Fajardo. "Estimating Surface Reflectance Properties of a Complex Scene Under Captured Natural Illumination," *Conditionally accepted to ACM Transactions on Graphics*, 2005.

[61] P. E. Debevec. "Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-Based Graphics with Illumination and High Dynamic Range Photography," in *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pp. 45–50, 1998.

[62] P. E. Debevec and J. Malik. "Recovering High Dynamic Range Radiance Maps from Photographs," in *Proceedings of SIGGRAPH 97*, Computer Graphics Proceedings, Annual Conference Series, pp. 369–378, August 1997.

[63] P. E. Debevec, C. J. Taylor, and J. Malik. "Modeling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-based Approach," in *Proceedings of SIGGRAPH 96*, Computer Graphics Proceedings, Annual Conference Series, pp. 11–20, August 1996.

[64] R. J. Deeley, N. Drasdo, and W. N. Charman. "A Simple Parametric Model of the Human Ocular Modulation Transfer Function," *Ophthalmology and Physiological Optics*, 11:91–93, 1991.

[65] A. M. Derrington, J. Krauskopf, and P. Lennie. "Chromatic Mechanisms in Lateral Geniculate Nucleus of Macaque," *Journal of Physiology*, 357:241–265, 1984.

[66] J. DiCarlo and B. Wandell. "Rendering High Dynamic Range Images," in *Proceedings of the SPIE Electronic Imaging 2000 Conference*, Vol. 3965, pp. 392–401, 2000.

[67] P. Didyk, R. Mantiuk, M. Hein, and H. P. Seidel. "Enhancement of Bright Video Features for HDR Displays," *Computer Graphics Forum*, 27(4):1265–1274, 2008.

[68] R. P. Dooley and M. I. Greenfield. "Measurements of Edge-Induced Visual Contrast and a Spatial-Frequency Interaction of the Cornsweet Illusion," *Journal of the Optical Society of America*, 67, pp. 761–765, 1977.

[69] J. E. Dowling. *The Retina: An Approachable Part of the Brain*. Cambridge, MA: Belknap Press, 1987.

[70] G. Downing. "Stitched HDRI," 2001, *www.gregdowning.com/HDRI/stitched/*.

[71] F. Drago, K. Myszkowski, T. Annen, and N. Chiba. "Adaptive Logarithmic Mapping for Displaying High Contrast Scenes," *Computer Graphics Forum*, 22(3):419–426, 2003.

[72] I. Drori, D. Cohen-Or, and H. Yeshurun. "Fragment-Based Image Completion," *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 22(3):303–312, 2003.

[73] F. Durand and J. Dorsey. "Interactive Tone Mapping," B. Peroche and H. Rushmeier (eds.) *11th Eurographics Workshop on Rendering*, Springer Verlag, pp. 219–230, 2000.

[74] F. Durand and J. Dorsey. "Fast Bilateral Filtering for the Display of High-Dynamic-Range Images," *ACM Transactions on Graphics*, 21(3):257–266, 2002.

[75] P. Dutré, P. Bekaert, and K. Bala. *Advanced Global Illumination*. Wellesley, MA: AK Peters, 2003.

[76] Eastman Kodak Company. Applied science fiction website, *http://www.asf.com/products/FPS/fpsfaqs.shtml*, accessed July 2004.

[77] F. Ebner and M. D. Fairchild. "Development and Testing of a Color Space (IPT) with Improved Hue Uniformity," in *IS&T Sixth Color Imaging Conference: Color Science, Systems and Applications*, pp. 8–13, 1998.

[78] G. Eckel and K. Jones. *OpenGL Performer Programmers Guide Chapter 18 – Light Points*. Silicon Graphics, 1997.

[79] S. R. Ellis, W. S. Kim, M. Tyler, M. W. McGreevy, and L. Stark. "Visual Enhancements for Perspective Displays: Perspective Parameters," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 297–305, 1985.

[80] P. G. Engeldrum. *Psychometric Scaling: A Toolkit for Imaging Systems Development*. Winchester, MA: Imcotek Press, 2000.

[81] M. D. Fairchild. "Revision of CIECAM97s for Practical Applications," *Color Research and Application*, 26:418–427, 2001.

[82] M. D. Fairchild. *Color Appearance Models*, 2nd ed., Chichester, UK: Wiley, 13(1):126–138, 2005.

[83] M. D. Fairchild and G. M. Johnson. "The iCAM Framework for Image Appearance, Image Differences, and Image Quality," *Journal of Electronic Imaging*, 13(1):1–34, 2004.

[84] M. D. Fairchild and G. M. Johnson. "Meet iCAM: An Image Color Appearance Model," in *IS&T/SID 10th Color Imaging Conference*, pp. 33–38, Scottsdale, 2002.

[85] M. Fajardo. "Monte Carlo Ray Tracing in Action," in *State of the Art in Monte Carlo Ray Tracing for Realistic Image Synthesis*, SIGGRAPH 2001 Course 29, August, 2001.

[86] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski. "Edge-Preserving Decompositions for Multi-Scale Tone and Detail Manipulation," *ACM Transactions on Graphics*, 27(3):67:1–67:10, 2008.

[87] H. Farid. "Blind Inverse Gamma Correction," *IEEE Transactions on Image Processing*, 10(10):1428–1433, 2001.

[88] R. Fattal. "Edge-Avoiding Wavelets and their Applications," *ACM Transactions on Graphics*, 28(3):22:1–10, 2009.

[89] R. Fattal, D. Lischinski, and M. Werman. "Gradient Domain High Dynamic Range Compression," *ACM Transactions on Graphics*, 21(3):249–256, 2002.

[90] E. A. Fedorovskaya, H. deRidder, and F. J. Blommaert. "Chroma Variations and Perceived Quality of Color Images of Natural Scenes," *Color Research and Application*, 22(2):96–110, 1997.

[91] J. A. Ferwerda. "Elements of Early Vision for Computer Graphics," *IEEE Computer Graphics and Applications*, 21(5):22–33, 2001.

[92] J. A. Ferwerda, S. Pattanaik, P. Shirley, and D. P. Greenberg. "A Model of Visual Adaptation for Realistic Image Synthesis," in H. Rushmeier (ed.),

SIGGRAPH 96 Conference Proceedings, Annual Conference Series, ACM SIGGRAPH, Addison Wesley, pp. 249–258, August 1996.

[93] J. A. Ferwerda, S. N. Pattanaik, P. S. Shirley, and D. P. Greenberg. "A Model of Visual Masking for Computer Graphics," in *Proceedings of SIGGRAPH 97*, Annual Conference Series, pp. 143–152, August 1997.

[94] G. D. Finlayson and S. Süsstrunk. "Color Ratios and Chromatic Adaptation," in *Proceedings of IS&T CGIV*, pp. 7–10, 2002.

[95] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer Graphics, Principles and Practice*, 2nd ed., Addison-Wesley, Reading MA, 1990.

[96] J. M. Foley. "Human Luminance Pattern-Vision Mechanisms: Masking Experiments Require a New Model," *Journal of the Optical Society of America A*, 11(6):1710–1719, 1994.

[97] J. Forrester, A. Dick, P. McMenamin, and W. Lee. *The Eye: Basic Sciences in Practice*. London: W B Saunders, 2001.

[98] G. R. Fowles. *Introduction to Modern Optics* (2nd ed.), New York: Dover Publications, 1975.

[99] O. Gallo, N. Gelfand, W. Chen, M. Tico, and K. Pulli. "Artifact-Free High Dynamic Range Imaging," in *IEEE International Conference on Computational Photography* (ICCP), April 2009.

[100] Y. Gao and Y. Wu. "Bit-Depth Scalability" in *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6*. Document JVT-V061, January 2007.

[101] A. Gardner, C. Tchou, T. Hawkins, and P. Debevec. "Linear Light Source Reflectometry," in *Proceedings of SIGGRAPH 2003*, Computer Graphics Proceedings, Annual Conference Series, pp. 335–342, 2003.

[102] A. Gardner, C. Tchou, T. Hawkins, and P. Debevec. "Linear Light Source Reflectometry," *ACM transactions on Graphics*, 22(3):749–758, 2003.

[103] W. S. Geisler. "Effects of Bleaching and Backgrounds on the Flash Response of the Cone System," *Journal of Physiology*, 312:413–434, 1981.

[104] A. S. Glassner. *Principles of Digital Image Synthesis*. San Fransisco, CA: Morgan Kaufmann, 1995.

[105] A. A. Gooch, S. C. Olsen, J. Tumblin, and B. Gooch. "Color2gray: Salience-Preserving Color Removal," *ACM Transactions on Graphics*, 24(3):634–639, 2005.

[106] J. W. Goodman. *Introduction To Fourier Optics*. Colorado, USA: Roberts & Co, 2005.

[107] N. Goodnight, R. Wang, C. Woolley, and G. Humphreys. "Interactive Time-Dependent Tone Mapping using Programmable Graphics Hardware," in *Proceedings of the 13th Eurographics workshop on Rendering*, pp. 26–37. Eurographics Association, 2003.

[108] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile. "Modeling the Interaction of Light Between Diffuse Surfaces," in *SIGGRAPH 84*, pp. 213–222, 1984.

[109] A. Goshtaby. "Fusion of Multi-Exposure Images," *Image and Vision Computing*, 23(6):611–618, 2005.

[110] A. Goshtaby and S. Nikolov. "Image Fusion: Advances in the State of the Art," *Information Fusion*, 8(2):114–118, 2007.

[111] N. Graham. *Visual Pattern Analyzer*. New York: Oxford University Press, 1989.

[112] N. Graham and D. C. Hood. "Modeling the Dynamics of Light Adaptation: The Merging of Two Traditions," *Vision Research*, 32:1373–1393, 1992.

[113] N. Greene. "Environment Mapping and Other Application of World Projections," *IEEE Computer Graphics and Applications*, 6(11):21–29, November 1986.

[114] S. G. de Groot and J. W. Gebhard. "Pupil Size as Determined by Adapting Luminance," *Journal of the Optical Society of America*, 42:492–495, 1952.

[115] T. Grosch. "Fast and Robust High Dynamic Range Image Generation with Camera and Object Movement," in *Vision, Modeling and Visualization, RWTH Aachen*, pp. 277–284, 2006.

[116] M. D. Grossberg and S. K. Nayar. "What Can Be Known about the Radiometric Response Function from Images?," in A. Heyden, G. Sparr, M. Nielsen, and P. Johansen (eds.), *7th European Conference on Computer Vision, Part IV*, Vol. 2353 of *Lecture Notes in Computer Science*, Springer, 2002.

[117] G. Guarnieri, L. Albani, and G. Ramponi. "Minimum-Error Splitting Algorithm for a Dual Layer LCD Display," *Journal of Display Technology*, 4(4):383–397, 2008.

[118] P. Haeberli. "Synthetic Lighting for Photography," January 1992, *www.sgi.com/grafica/synth/index.html*.

[119] T. R. Halfhill. "Parallel Processing with CUDA," *Microprocessor Report*, 1/28/08-01, 2008.

[120] R. Hall. *Illumination and Color in Computer Generated Imagery*. New York, NY, Springer-Verlag, 1989.

[121] V. Havran, M. Smyk, G. Krawczyk, K. Myszkowski, and H.-P. Seidel. "Interactive System for Dynamic Scene Lighting using Captured Video Environment Maps," in *Rendering Techniques 2005: Eurographics Symposium on Rendering*, pp. 31–42,311, Konstanz, Germany: Eurographics, 2005.

[122] E. Hecht. *Optics,* 2nd ed., Reading MA: Addison-Wesley, 1987.

[123] P. Heckbert. "Color Image Quantization for Frame Buffer Display," in *SIGGRAPH'84: Proceedings of the 9th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 297–307, ACM Press, July 1982.

[124] W. Heidrich and H.-P. Seidel. "Realistic, Hardware-accelerated Shading and Lighting," in *Proceedings of SIGGRAPH 99*, pp. 171–178, August 1999.

[125] R. D. Hersch. "Spectral Prediction Model for Color Prints on Paper with Fluorescent Additives," *Applied Optics*, 47(36):6710-6722, 2008.

[126] R. D. Hersch, P. Donze, and S. Chosson. "Color Images Visible Under UV Light," *ACM Transactions on Graphics*, 26(3):75, 2007.

[127] B. Hoefflinger. *High-Dynamic-Range (HDR) Vision*. New York, NY, Springer-Verlag, 2006.

[128] M. Hogan, J. Alvarado, and J. Weddell. *Histology of the Human Eye*. Philadelphia, PA: W. B. Saunders, 1971.

[129] D. C. Hood and M. A. Finkelstein. "Comparison of Changes in Sensitivity and Sensation: Implications for the Response-Intensity Function of the Human Photopic System," *Journal of Experimental Psychology: Human Perceptual Performance*, 5:391–405, 1979.

[130] D. C. Hood and M. A. Finkelstein. "Sensitivity to Light," in K. R. Boff, L. R. Kaufman, and J. P. Thomas (eds.), *Handbook of Perception and Human Performance*, New York: Wiley, pp. 1–66, 1986.

[131] D. C. Hood, M. A. Finkelstein, and E. Buckingham. "Psychophysical Tests of Models of the Response Function," *Vision Research*, 19:401–406, 1979.

[132] B. K. P. Horn. "Determining Lightness from an Image," *Computer Graphics and Image Processing*, 3:277–299, 1974.

[133] R. W. G. Hunt and M. R. Luo. "The Structure of the CIECAM97 Colour Appearance Model (CIECAM97s)," in *CIE Expert Symposium '97*, Scottsdale, 1997.

[134] R. W. G. Hunt. *The Reproduction of Color*, 6th ed., Chichester, UK: Wiley and Sons.

[135] P. Irawan, J. A. Ferwerda, and S. R. Marschner. "Perceptually-Based Tone Mapping of High Dynamic Range Image Streams," in *Eurographics Symposium on Rendering*, pp. 231–242, Eurographics Association, 2005.

[136] ISO-IEC 14496-2. *Information Technology: Coding of Audio-Visual Objects, Part 2: Visual*. Geneva, Switzerland: International Organization for Standardization, 1999.

[137] ISO/IEC 14496-10. *Information Technology: Coding of Audio-Visual Objects, Part 10: Advanced Video Coding*. Geneva, Switzerland: International Organization for Standardization, 2005.

[138] ITU (International Telecommunication Union), Geneva. *ITU-R Recommendation BT.709, Basic Parameter Values for the HDTV Standard for the Studio and for International Programme Exchange*, 1990. Formerly CCIR Rec. 709, Geneva: ITU.

[139] K. Jacobs, G. Ward, and C. Loscos. "Automatic HDRI Generation of Dynamic Environments," in *SIGGRAPH '05: ACM SIGGRAPH 2005 Sketches*, p. 43, New York: ACM, 2005.

[140] R. Janssen. *Computational Image Quality*. Bellingham, WA, United States: SPIE Press, 2001.

[141] H. W. Jensen. *Realistic Image Synthesis Using Photon Mapping*. Natick, MA: AK Peters, 2001.

[142] D. J. Jobson, Z. Rahman, and G. A. Woodell. "Retinex Image Processing: Improved Fidelity to Direct Visual Observation," in *Proceedings of the IS&T Fourth Color Imaging Conference: Color Science, Systems, and Applications*, Vol. 4, pp. 124–125, 1995.

[143] G. M. Johnson and M. D. Fairchild. "Rendering HDR Images," in *IS&T/SID 11th Color Imaging Conference*, pp. 36–41, Scottsdale, 2003.

[144] F. Kainz, R. Bogart, and D. Hess. "The Openexr Image File Format," in *SIGGRAPH Technical Sketches*, 2003. See also: *www.OpenEXR.com*.

[145] J. T. Kajiya. "The Rendering Equation," in *Computer Graphics (Proceedings of SIGGRAPH 86)*, Vol. 20, pp. 143–150, 1986.

[146] M. Kakimoto, K. Matsuoka, T. Nishita, T. Naemura, and H. Harashima. "Glare Generation Based on Wave Optics," in *Proceedings of the 12th Pacific Conference on Computer Graphics and Applications (PG'04)*, pp. 133–142, Washington, DC: IEEE Computer Society, 2004.

[147] S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. "High Dynamic Range Video," *ACM Transactions on Graphics*, 22(3):319–325, 2003.

[148] N. Katoh and K. Nakabayashi. "Applying Mixed Adaptation to Various Chromatic Adaptation Transformation (CAT) Models," in *Proceedings PICS Conference*, pp. 299–305, 2001.

[149] M. Kawase. "Real-time High Dynamic Range Image-based Lighting," 2003, *www.daionet.gr.jp/~masa/rthdribl/*.

[150] M. Kawase. "Practical Implementation of High Dynamic Range Rendering," in *Game Developers Conference*, 2005.

[151] M. S. Keil. "Gradient Representations and the Perception of Luminosity," *Vision Research*, 47(27):3360–3372, 2007.

[152] D. H. Kelly. "Motion and Vision 1: Stabilized Images of Stationary Gratings," *Journal of the Optical Society of America A*, 69(9):1266–1274, 1979.

[153] D. H. Kelly. "Motion and Vision 2: Stabilized Spatio-Temporal Threshold Surface," *Journal of the Optical Society of America*, 69(10):1340–1349, 1979.

[154] D. H. Kelly. "Spatiotemporal Variation of Chromatic and Achromatic Contrast Thresholds," *Journal of the Optical Society of America*, 73(6):742–750, 1983.

[155] E. A. Khan, E. Reinhard, R. Fleming, and H. Bulthoff. "Image-Based Material Editing," *ACM Transactions on Graphics*, 25(3):654–663, 2006.

[156] M. H. Kim, T. Weyrich, and J. Kautz. "Modeling Human Color Perception under Extended Luminance Levels," *ACM Transactions on Graphics*, 28(3):27, 2009.

[157] F. Kingdom and B. Moulden. "Border Effects on Brightness: A Review of Findings, Models and Issues," *Spatial Vision*, 3(4):225–262, 1988.

[158] J. Kleinschmidt and J. E. Dowling. "Intracellular Recordings from Gecko Photoreceptors during Light and Dark Adaptation," *Journal of General Physiology*, 66:617–648, 1975.

[159] C. Kolb, D. Mitchell, and P. Hanrahan. "A Realistic Camera Model for Computer Graphics," in *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 317–324, 1995.

[160] T. Kollig and A. Keller. "Efficient Illumination by High Dynamic Range Images," in *Eurographics Symposium on Rendering: 14th Eurographics Workshop on Rendering*, pp. 45–51, 2003.

[161] G. Krawczyk, R. Mantiuk, K. Myszkowski, and H.-P. Seidel. "Lightness Perception Inspired Tone Mapping," in *Proceedings of the 1st ACM Symposium on Applied Perception in Graphics and Visualization*, pp. 172, 2004.

[162] G. Krawczyk, R. Mantiuk, D. Zdrojewska, and H.-P. Seidel. "Brightness Adjustment for HDR and Tone Mapped Images," in *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications (PG'07)*, 2007.

[163] G. Krawczyk, K. Myszkowski, and H.-P. Seidel. "Lightness Perception in Tone Reproduction for High Dynamic Range Images," *Computer Graphics Forum*, 24(3):635–645, 2005.

[164] G. Krawczyk, K. Myszkowski, and H.-P. Seidel. "Perceptual Effects in Real-Time Tone Mapping," in *SCCG '05: Proceedings of the 21st Spring Conference on Computer Graphics*, pp. 195–202, 2005.

[165] G. Krawczyk, K. Myszkowski, and H.-P. Seidel. "Computational Model of Lightness Perception in High Dynamic Range Imaging," in *Proceedings of IS&T/SPIE Human Vision and Electronic Imaging*, Vol. 6057, 2006.

[166] G. Krawczyk, K. Myszkowski, and H.-P. Seidel. "Contrast Restoration by Adaptive Countershading," in *The European Association for Computer Graphics Annual Conference EUROGRAPHICS 2007*, Vol. 26 of *Computer Graphics Forum*, Blackwell, 2007.

[167] J. von Kries. "Chromatic Adaptation," in D. L. MacAdam (ed.), *Sources of Color Science*, pp. 120–126, Cambridge, MA: MIT Press, 1902, reprinted 1970.

[168] J. Kuang, G. M. Johnson, and M. D. Fairchild. "iCAM06: A Refined Image Appearance Model for HDR Image Rendering," *Journal of Visual Communication and Image Representation*, 18(5):406–414, 2007.

[169] T. Kunkel and E. Reinhard. "A Neurophysiology-Inspired Steady-State Color Appearance Model," *Journal of the Optical Society of America A*, 26(4):776–782, 2009.

[170] E. P. F. Lafortune, S.-C. Foo, K. E. Torrance, and D. P. Greenberg. "Non-linear Approximation of Reflectance Functions," in *Proceedings of SIGGRAPH 97*, pp. 117–126, 1997.

[171] Y. K. Lai and C.-C. J. Kuo. "A Haar Wavelet Approach to Compressed Image Quality Measurement," *Journal of Visual Communication and Image Representation*, 11(1):17–40, 2000.

[172] J. F. Lalonde, A. A. Efros, and S. Narasimhan. "Webcam Clip Art: Appearance and Illuminant Transfer From Time-Lapse Sequences," *ACM Transactions on Graphics*, 28(5):131, 2009.

[173] H. Landis. "Production-ready Global Illumination," *Course notes for SIGGRAPH 2002 Course 16, "RenderMan in Production,"* 2002.

[174] G. W. Larson. "Logluv Encoding for Full-Gamut, High Dynamic Range Images," *Journal of Graphics Tools*, 3(1):15–31, 1998.

[175] G. W. Larson. "Overcoming Gamut and Dynamic Range Limitations in Digital Images," in *Proceedings of IS&T 6th Color Imaging Conference*, 1998.

[176] G. W. Larson, H. Rushmeier, and C. Piatko. "A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes," *IEEE Transactions on Visualization and Computer Graphics*, 3(4):291–306, October–December 1997. ISSN 1077-2626.

[177] G. W. Larson and R. A. Shakespeare. *Rendering with Radiance*. Morgan Kaufmann, San Francisco, 1998.

[178] J. Lawrence, S. Rusinkiewicz, and R. Ramamoorthi. "Efficient BRDF Importance Sampling Using a Factored Representation," in *ACM Transactions on Graphics (SIGGRAPH 2004)*, August 2004.

[179] P. Ledda, A. Chalmers, and H. Seetzen. "HDR Displays: A Validation Against Reality," in *International Conference on Systems, Man and Cybernetics*, The Hague, The Netherlands, pp. 2777–2782, October 2004.

[180] P. Ledda, A. Chalmers, T. Troscianko, and H. Seetzen. "Evaluation of Tone Mapping Operators Using a High Dynamic Range Display," *ACM Transactions on Graphics*, 24(3):640–648, 2005.

[181] P. Ledda, L. P. Santos, and A. Chalmers. "A Local Model of Eye Adaptation for High Dynamic Range Imagines," in *Proceedings of AFRIGRAPH*, pp. 151–160, 2004.

[182] G. E. Legge and J. M. Foley. "Contrast Masking in Human Vision," *Journal of the Optical Society of America*, 70(12):1458–1471, 1980.

[183] B. Li, G. W. Meyer, and R. V. Klassen. "A Comparison of Two Image Quality Models," in *Human Vision and Electronic Imaging III*, Vol. 3299, pp. 98–109. SPIE, 1998.

[184] C. Li, M. R. Luo, B. Rigg, and R. W. G. Hunt. "CMC 2000 Chromatic Adaptation Transform: CMCCAT2000," *Color Research & Application*, 27:49–58, 2002.

[185] C. Li, M. R. Luo, R. W. G. Hunt, N. Moroney, M. D. Fairchild, and T. Newman. "The Performance of CIECAM02," in *IS&T/SID 10th Color Imaging Conference*, pp. 28–32, Scottsdale, 2002.

[186] Y. Li, L. Sharan, and E. H. Adelson. "Compressing and Companding High Dynamic Range Images with Subband Architectures," *ACM Transactions on Graphics*, 24(3):836–844, 2005.

[187] T. M. Lillesand and R. W. Kiefer and J. Chipman, *Remote Sensing and Image Interpretation 6th ed*. New York: John Wiley & Sons, 1994.

[188] S. Lin, J. Gu, S. Yamazaki, and H.-Y. Shum. "Radiometric Calibration from a Single Image," *Conference on Computer Vision and Pattern Recognition (CVPR'04)*, 2:938–945, 2004.

[189] W. S. Lin, Y. L. Gai, and A. A. Kassim. "Perceptual Impact of Edge Sharpness in Images," *Vision, Image and Signal Processing, IEE Proceedings*, 152(2):215–223, 2006.

[190] B. Lindbloom. *www.brucelindbloom.com*.

[191] S. Liu, W.-S. Kim, and A. Vetro. "Bit-Depth Scalable Coding for High Dynamic Range Video," in W. A. Pearlman, J. W. Woods and L. Lu (eds.), *Visual Communications and Image Processing*, Vol. 6822, pp. 1–10, San Jose, CA: SPIE, 2008.

[192] X. Liu, P.-P. Sloan, H.-Y. Shum, and J. Snyder. "All-frequency Precomputed Radiance Transfer for Glossy Objects," in *Rendering Techniques 2004: 15th Eurographics Workshop on Rendering*, pp. 337–344, June 2004.

[193] M. Livingstone. *Vision and Art: The Biology of Seeing*. New York, Harry N. Abrams, 2002.

[194] J. Lubin. "Vision Models for Target Detection and Recognition," in E. Peli (ed.), *A Visual Discrimination Model for Imaging System Design and Evaluation*, pp. 245–283. Singapore: World Scientific, 1995.

[195] J. Lubin and A. P. Pica. "A Non-Uniform Quantizer Matched to the Human Visual Performance," *Society of Information Display International Symposium Technical Digest of Papers*, 22:619–622, 1991.

[196] B. D. Lucas and T. Kanade. "An Iterative Image Registration Technique with an Application in Stereo Vision," in *Seventh International Joint Conference on Artificial Intelligence (IJCAI-81)*, pp. 674–679, 1981.

[197] T. Luft, C. Colditz, and O. Deussen. "Image Enhancement by Unsharp Masking the Depth Buffer," *ACM Transactions on Graphics*, 25(3):1206–1213, 2006.

[198] M. R. Luo, A. A. Clark, P. A. Rhodes, A. Schappo, S. A. R. Scrivner, and C. J. Tait. "Quantifying Colour Appearance Part I: LUTCHI Colour Appearance Data," *Color Research and Application*, 16:166–180, 1991.

[199] M. R. Luo, A. A. Clark, P. A. Rhodes, A. Schappo, S. A. R. Scrivner, and C. J. Tait. "Quantifying Colour Appearance Part II: Testing Colour Models Performance using LUTCHI Color Appearance Data," *Color Research and Application*, 16:181–197, 1991.

[200] J. B. MacQueen. "Some Methods for Classification and Analysis of Multivariate Observations," in *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, pp. 281–297, Berkeley: University of California Press, 1997.

[201] S. Mann and R. W. Picard. "Being Undigital with Digital Cameras: Extending Dynamic Range by Combining Differently Exposed Pictures," in *IS&T's 48th Annual Conference*, pp. 422–428, Springfield, VA, 1995.

[202] R. Mantiuk, S. Daly, and L. Kerofsky. "Display Adaptive Tone Mapping," *ACM Transactions on Graphics*, 27(3):68, 2008.

[203] R. Mantiuk, S. Daly, K. Myszkowski, and H.-P. Seidel. "Predicting Visible Differences in High Dynamic Range Images – Model and its Calibration," in *Human Vision and Electronic Imaging X*, Vol. 5666 of *SPIE Proceedings Series*, pp. 204–214, 2005.

[204] R. Mantiuk, A. Efremov, K. Myszkowski, and H.-P. Seidel. "Backward Compatible High Dynamic Range MPEG Video Compression," *ACM Transactions on Graphics*, 25(3):713–723, 2006.

[205] R. Mantiuk, G. Krawczyk, K. Myszkowski, and H.-P. Seidel. "Perception-motivated High Dynamic Range Video Encoding," *ACM Transactions on Graphics*, 23(3):730–738, 2004.

[206] R. Mantiuk, G. Krawczyk, K. Myszkowski, and H.-P. Seidel. "High Dynamic Range Image and Video Compression – Fidelity Matching Human Visual Performance," in *Proceedings of the International Conference on Image Processing, ICIP 2007*, San Antonio, TX, pp. 9–12, 2007.

[207] R. Mantiuk, K. Myszkowski, and H.-P. Seidel. "Lossy Compression of High Dynamic Range Images and Video," in *Proceedings of Human Vision and Electronic Imaging XI*, Vol. 6057 of *Proceedings of SPIE*, pp. 60570V, San Jose, CA: SPIE, 2006.

[208] R. Mantiuk, K. Myszkowski, and H.-P. Seidel. "A Perceptual Framework for Contrast Processing of High Dynamic Range Images," *ACM Transactions on Applied Perception*, 3(3):286–308, 2006.

[209] S. Marcelja. "Mathematical Description of the Responses of Simple Cortical Cells," *Journal of the Optical Society of America*, 70:1297–1300, 1980.

[210] D. H. Marimont and B. A. Wandell. Matching Colour Images: The Effects of Axial Chromatic Aberration," *Journal of the Optical Society of America A*, 11(12):3113–3122, 1994.

[211] W. R. Mark, R. S. Glanville, K. Akeley, and M. J. Kilgard. "Cg: A System for Programming Graphics Hardware in a C-like Language," *ACM Transactions on Graphics*, 22(3):896–907, 2003.

[212] T. L. Martzall. "Simultaneous Raster and Calligraphic CRT Projection System for Flight Simulation," in *Proceedings of SPIE (Projection Displays)*, Vol. 1910, pp. 292–299, 1993.

[213] B. Masia, S. Agustin, R. W. Fleming, O. Sorkine, and D. Gutierrez. "Evaluation of Reverse Tone Mapping Through Varying Exposure Conditions," *ACM Transactions on Graphics*, 28(5):160, 2009.

[214] D. K. McAllister. "A Generalized Surface Appearance Representation for Computer Graphics," Ph.D. thesis, University of North Carolina at Chapel Hill, 2002.

[215] J. McCann and A. Rizzi. "Veiling Glare: The Dynamic Range Limit of HDR Images," in *Human Vision and Electronic Imaging XII*, Vol. 6492 of *SPIE Proceedings Series*, 2007.

[216] R. McDonald and K. J. Smith. "CIE94 — A New Colour-Difference Formula," *Society Dyers Colegia*, 11:376–379, 1995.

[217] T. Mertens, J. Kautz, and F. van Reeth. "Exposure Fusion," in *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, pp. 382–390, 2007.

[218] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. "Equations of State Calculations by Fast Computing Machines," *Journal of Chemical Physics*, 21:1087–1091, 1953.

[219] L. Meylan, D. Alleysson, and S. Süsstrunk. "Model of Retinal Local Adaptation for the Tone Mapping of Color Filter Array Images," *Journal of the Optical Society of America A*, 24(9):2807–2816, 2007.

[220] L. Meylan, S. Daly, and S. Susstrunk. "The Reproduction of Specular Highlights on High Dynamic Range Displays," in *Proceedings of the 14th Color Imaging Conference*, 2006.

[221] L. Meylan, S. Daly, and S. Susstrunk. "Tone Mapping for High Dynamic Range Displays," in *Human Vision and Electronic Imaging XII*, Vol. 6492, SPIE, 2007.

[222] G. S. Miller and C. R. Hoffman. "Illumination and Reflection Maps: Simulated Objects in Simulated and Real Environments," in *SIGGRAPH 84 Course Notes for Advanced Computer Graphics Animation*, July 1984.

[223] J. Mitchell, J. Isidoro, and A. Vlachos. "ATI Radeon 9700 Real-time Demo of Rendering with Natural Light," 2002, *www.ati.com/developer/demos/R9700.html*.

[224] T. Mitsunaga and S. K. Nayar. "Radiometric Self Calibration," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Fort Collins, 374–380, 1999.

[225] P. Moon and D. E. Spencer. "Visual Data Applied to Lighting Design," *Journal of the Optical Society of America*, 34(10):605–617, 1944.

[226] N. Moroney, M. D. Fairchild, R. W. G. Hunt, C. J. Li, M. R. Luo, and T. Newman. "The CIECAM02 Color Appearance Model," in *IS&T 10th Color Imaging Conference*, pp. 23–27, Scottsdale, 2002.

[227] N. Moroney. "Usage Guidelines for CIECAM97s," in *Proceedings of the Conference on Image Processing, Image Quality, Image Capture Systems (PICS-00)*, pp. 164–168, Springfield, Virginia, March 26–29, 2000. Society for Imaging Science and Technologie.

[228] N. Moroney and I. Tastl. "A Comparison of Retinex and iCAM for Scene Rendering," *Journal of Electronic Imaging*, 13(1):139–145, 2004.

[229] K. T. Mullen. "The Contrast Sensitivity of Human Color Vision to Red-Green and Blue-Yellow Chromatic Gratings," *Journal of Psychology*, 359:381–400, 1985.

[230] J. Munkberg, P. Clarberg, J. Hasselgren, and T. Akenine-Möller. "Practical HDR Texture Compression," *Computer Graphics Forum*, 27(6):1664–1676, 2008.

[231] K. Myszkowski. "The Visible Differences Predictor: Applications to Global Illumination Problems," in G. Drettakis and N. Max (eds.), *Rendering Techniques '98*, pp. 223–236. Springer, 1998.

[232] K. Myszkowski, R. Mantiuk, and G. Krawczyk. *High Dynamic Range Video*. Synthesis Digital Library of Engineering and Computer Science. San Rafael: Morgan & Claypool Publishers, 2008.

[233] J. Nachmias. "On the Psychometric Function for Contrast Detection," *Vision Research*, 21:215–223, 1981.

[234] K. I. Naka and W. A. H. Rushton. "S-Potentials from Luminosity Units in the Retina of Fish (Cyprinidae)," *Journal of Physiology*, 185:587–599, 1966.

[235] E. Nakamae, K. Kaneda, T. Okamoto, and T. Nishita. "A Lighting Model Aiming at Drive Simulators,". in *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pp. 395–404, New York ACM, 1990.

[236] S. K. Nayar. "Catadioptric Omnidirectional Camera," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 482–488, Puerto Rico, June 1997.

[237] F. van Nes and M. Bouman. "Spatial Modulation for Gray and Color Images," *Journal of the Optical Society of America*, 57:401–406, 1967.

[238] F. Neycenssac. "Contrast Enhancement using the Laplacian-of-a-Gaussian Filter," *CVGIP: Graphical Models and Image Processing*, 55(6):447–463, 1993.

[239] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. "Geometric Considerations and Nomenclature for Reflectance," *National Bureau of Standards Monograph 160*, October 1977.

[240] R. Ng, R. Ramamoorthi, and P. Hanrahan. "All-frequency Shadows Using Non-linear Wavelet Lighting Approximation," *ACM Transactions on Graphics*, 22(3):376–381, July 2003.

[241] A. V. Oppenheim, R. Schafer, and T. Stockham. "Nonlinear Filtering of Multiplied and Convolved Signals," *Proceedings of the IEEE*, 56(8):1264–1291, 1968.

[242] V. Ostromoukhov, C. Donohue, and P.-M. Jodoin. "Fast Hierarchical Importance Sampling with Blue Noise Properties," *ACM Transactions on Graphics*, 23(3):488–495, August 2004.

[243] S. E. Palmer. *Vision Science: Photons to Phenomenology*. Cambridge, MA: MIT Press, 1999.

[244] S. Paris and F. Durand. "A Fast Approximation of the Bilateral Filter using a Signal Processing Approach," *International Journal of Computer Vision*, 81(1): 24–52, 2007.

[245] D. Pascale. "A Review of RGB Color Spaces," Technical report, The BabelColor Company, 2003.

[246] S. N. Pattanaik, J. A. Ferwerda, M. D. Fairchild, and D. P. Greenberg. "A Multiscale Model of Adaptation and Spatial Vision for Realistic Image Display," in *SIGGRAPH 98 Conference Proceedings*, pp. 287–298, 1998.

[247] S. N. Pattanaik, J. Tumblin, H. Yee, and D. P. Greenberg. "Time-Dependent Visual Adaptation for Fast Realistic Display," in *SIGGRAPH 2000 Conference Proceedings*, pp. 47–54, 2000.

[248] S. N. Pattanaik, J. E. Tumblin, H. Yee, and D. P. Greenberg. "Time-Dependent Visual Adaptation for Fast Realistic Image Display," in *Proceedings of ACM SIGGRAPH 2000*, pp. 47–54, 2000.

[249] S. N. Pattanaik and H. Yee. "Adaptive Gain Control for High Dynamic Range Image Display," in *Proceedings of Spring Conference on Computer Graphics (SCCG2002)*, pp. 24–27, Budmerice, Slovak Republic, 2002.

[250] E. Peli. "Contrast in Complex Images," *Journal of the Optical Society of America A*, 7(10):2032–2040, 1990.

[251] P. Perez, M. Gangnet, and A. Blake. "Poisson Image Editing," *ACM Transactions on Graphics*, 22(3):313–318, 2003.

[252] M. Pharr and G. Humphreys. "Improved Infinite Area Light Source Sampling," 2004, *http://pbrt.org/plugins.php*.

[253] M. Pharr and G. Humphreys. *Physically Based Rendering: From Theory to Implementation*. San Francisco: Morgan Kaufmann, 2004.

[254] B. Phong. "Illumination for Computer Generated Pictures," *Communications of the ACM*, 18(6), September 1975.

[255] A. Polesel, G. Ramponi, and V. Mathews. "Image Enhancement via Adaptive Unsharp Masking," *IEEE Transactions on Image Processing*, 9:505–510, 2000.

[256] C. Poynton. *Digital Video and HDTV: Algorithms and Interfaces*. San Francisco: Morgan Kaufmann Publishers, 2003.

[257] W. K. Pratt. *Digital Image Processing*, 2nd ed., New York: John Wiley & Sons, 1991.

[258] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed., Cambridge, UK: Cambridge University Press, 1992.

[259] D. Purves, A. Shimpi, and B. R. Lotto. "An Empirical Explanation of the Cornsweet Effect," *Journal of Neuroscience*, 19(19):8542–8551, 1999.

[260] Z. Rahman, D. J. Jobson, and G. A. Woodell. "A Multiscale Retinex for Color Rendition and Dynamic Range Compression," in *SPIE Proceedings: Applications of Digital Image Processing XIX*, Vol. 2847, 1996.

[261] Z. Rahman, G. A. Woodell, and D. J. Jobson. "A Comparison of the Multiscale Retinex with Other Image Enhancement Techniques," in *IS&T's 50th Annual Conference: A Celebration of All Imaging*, Vol. 50, pp. 426–431, 1997.

[262] R. Ramamoorthi and P. Hanrahan. "Frequency Space Environment Map Rendering," *ACM Transactions on Graphics*, 21(3):517–526, July 2002.

[263] S. Raman and S. Chaudhuri. "Bilateral Filter-Based Compositing for Variable Exposure Photography," in *Proceedings of Eurographics '09, Short Papers*, Eurographics Association, 2009.

[264] M. Ramasubramanian, S. N. Pattanaik, and D. P. Greenberg. "A Perceptually-Based Physical Error Metric for Realistic Image Synthesis," in *Proceedings of SIGGRAPH 99*, Annual Conference Series, pp. 73–82, 1999.

[265] G. Ramponi, N. Strobel, S. K. Mitra, and T.-H. Yu. "Nonlinear Unsharp Masking Methods for Image Contrast Enhancement," *Journal of Electronic Imaging*, 5:353–366, 1996.

[266] R. Raskar, A. Agrawal, C. A. Wilson, and A. Veeraraghavan. "Glare Aware Photography: 4D Ray Sampling for Reducing Glare Effects of Camera Lenses," *ACM Transactions on Graphics*, 27(3):56, 2008.

[267] F. Ratliff. "Contour and Contrast," *Proceedings of the American Philosophical Society*, 115(2):150–163, 1971.

[268] M. S. Rea and I. G. Jeffrey. "A New Luminance and Image Analysis System for Lighting and Vision: I. Equipment and Calibration," *Journal of the Illuminating Engineering Society*, 9(1):64–72, 1990.

[269] M. S. Rea (ed.), *The IESNA Lighting Handbook: Reference and Application*. The Illuminating Engineering Society of North America, New York, NY, USA, 2000.

[270] E. Reinhard. "Parameter Estimation for Photographic Tone Reproduction," *Journal of Graphics Tools*, 7(1):45–51, 2003.

[271] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley. "Color Transfer between Images," *IEEE Computer Graphics and Applications*, 21:34–41, 2001.

[272] E. Reinhard and K. Devlin. "Dynamic Range Reduction Inspired by Photoreceptor Physiology," *IEEE Transactions on Visualization and Computer Graphics*, 11(1):13–24, 2005.

[273] E. Reinhard, T. Kunkel, Y. Marion, J. Brouillat, R. Cozot, and K. Bouatouch. "Image Display Algorithms for High and Low Dynamic Range Display Devices," *Journal of the Society for Information Display*, 15(12):997–1014, 2007.

[274] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda. "Photographic Tone Reproduction for Digital Images," *ACM Transactions on Graphics*, 21(3):267–276, 2002.

[275] A. G. Rempel, M. Trentacoste, H. Seetzen, H. D. Young, W. Heidrich, L. Whitehead, and G. Ward. "Ldr2Hdr: On-the-Fly Reverse Tone Mapping of Legacy Video and Photographs," *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 26(3), 2007. Article 39.

[276] T. Ritschel, M. Ihrke, J. R. Frisvad, J. Coppens, K. Myszkowski, and H.-P. Seidel. "Temporal Glare: Real-Time Dynamic Simulation of the Scattering in the Human Eye," *Computer Graphics Forum (Proceedings of EUROGRAPHICS 2009)*, 28(3):183–192, 2009.

[277] T. Ritschel, K. Smith, M. Ihrke, T. Grosch, K. Myszkowski, and H.-P. Seidel. "3D Unsharp Masking for Scene Coherent Enhancement," *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 27(3):90, 2008.

[278] P. Rokita. "A Model for Rendering High Intensity Lights," *Computers & Graphics*, 17(4):431–437, 1993.

[279] R. Rosenholtz and A. B. Watson. "Perceptual Adaptive JPEG Coding," in *IEEE International Conference on Image Processing*, pp. 901–904, 1996.

[280] D. L. Ruderman, T. W. Cronin, and C.-C. Chiao. "Statistics of Cone Responses to Natural Images: Implications for Visual Coding," *Journal of the Optical Society of America A*, 15(8):2036–2045, 1998.

[281] W. A. H. Rushton and D. I. A. MacLeod. "The Equivalent Background of Bleaching," *Perception*, 15:689–703, 1986.

[282] C. Schlick. "Quantization Techniques for the Visualization of High Dynamic Range Pictures," in P. Shirley, G. Sakas, and S. Müller (eds.), *Photorealistic Rendering Techniques*, pp. 7–20, New York: Springer-Verlag, 1994.

[283] H. Schwarz, D. Marpe, and T. Wiegand. "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1103–1120, 2007.

[284] H. Seetzen, H. Li, L. Ye, W. Heidrich, and L. Whitehead. "Observations of Luminance, Contrast and Amplitude Resolution of Displays," in *Society for Information Display (SID) '06*, pp. 1229–1233, 2006.

[285] H. Seetzen, L. Whitehead, and G. Ward. "A High Dynamic Range Display using Low and High Resolution Modulators," in *Society for Information Display International Symposium Digest of Technical Papers*, pp. 1450–1453, 2003.

[286] H. Seetzen. "High Dynamic Range Display and Projection Systems," Ph.D. thesis, University of British Columbia, 2009.

[287] H. Seetzen, W. Heidrich, W. Stuerzlinger, G. Ward, L. Whitehead, M. Trentacoste, A. Ghosh, and A. Vorozcovs. "High Dynamic Range Display Systems," *ACM Transactions on Graphics*, 23(3):760–768, 2004.

[288] H. Seetzen, L. A. Whitehead, and G. Ward. "A High Dynamic Range Display using Low and High Resolution Modulators," in *The Society for Information Display International Symposium*, 34(1):1450–1453, 2003.

[289] C. A. Segall. "Scalable Coding of High Dynamic Range Video," in *Proceedings of the 2007 International Conference on Image Processing,* San Antonio, TX, pp. 1–4, 2007.

[290] M. I. Sezan, K. L. Yip, and S. Daly. "Uniform Perceptual Quantization: Applications to Digital Radiography," *IEEE Transactions on Systems, Man, and Cybernetics*, 17(4):622–634, 1987.

[291] G. Sharma, W. Wu, and E. N. Dalal. "The CIEDE2000 Color-Difference Formula: Implementation Notes, Supplementary Test Data, and Mathematical Observations," *Color Research and Application*, 30(1):21–30, 2005.

[292] P. Shirley, S. Marshner, M. Ashikhmin, M. Gleicher, N. Hoffman, G. Johnson, T. Munzner, E. Reinhard, K. Sung, W. Thompson, P. Willemsen and B. Wyvill (3 ed.). *Fundamentals of Computer Graphics*. Natick, MA: AK Peters, 3: 2009.

[293] F. X. Sillion and C. Puech. *Radiosity and Global Illumination*. San Francisco, CA: Morgan Kaufmann Publishers, Inc., 1994.

[294] G. C. Simpson. "Ocular Haloes and Coronas," *British Journal of Ophthalmology*, 37(8):450–486, 1953.

[295] P.-P. Sloan, J. Kautz, and J. Snyder. "Precomputed Radiance Transfer for Real-time Rendering in Dynamic, Low-frequency Lighting Environments," *ACM Transactions on Graphics*, 21(3):527–536, July 2002.

[296] K. Smith. "Contours and Contrast," Ph.D. thesis, MPI Informatik, Saarbruecken, Germany, 2008.

[297] K. Smith, G. Krawczyk, K. Myszkowski, and H.-P. Seidel. "Beyond Tone Mapping: Enhanced Depiction of Tone Mapped HDR images," *Computer Graphics Forum*, 25(3):427–438, 2006.

[298] K. Smith, P.-E. Landes, J. Thollot, and K. Myszkowski. "Apparent Greyscale: A Simple and Fast Conversion to Perceptually Accurate Images and Video," *Computer Graphics Forum*, 27(2):193–200, 2008.

[299] S. M. Smith and J. M. Brady. "SUSAN1/m A New Approach to Low Level Image Processing," *International Journal of Computer Vision*, 23(1):45–78, 1997.

[300] B. Smith and G. Meyer. "Simulating Interference Phenomena in Realistic Image Synthesis," in *Proceedings of the First Eurographic Workshop on Rendering*, pp. 185–194, 1990.

[301] B. Smith and L. Rowe. "Compressed Domain Processing of JPEG-encoded Images," *Real-Time Imaging*, 2(2):3–17, 1996.

[302] K. E. Spaulding, G. J. Woolfe, and R. L. Joshi. "Using a Residual Image to Extend the Color Gamut and Dynamic Range of an SRGB Image," in *Proceedings of IS&T PICS Conference*, pp. 307–314, 2003.

[303] G. Spencer, P. Shirley, K. Zimmerman, and D. P. Greenberg. "Physically-Based Glare Effects for Digital Images," in *Proceedings of SIGGRAPH 99*, Vol. 29 of *Computer Graphics Proceedings, Annual Conference Series*, pp. 325–334, 1995.

[304] L. Spillmann and J. S. Werner, eds. *Visual Perception: The Neurological Foundations*. San Diego, CA: Academic Press, 1990.

[305] S. S. Stevens and J. C. Stevens. "Brightness Function: Parametric Effects of Adaptation and Contrast," *Journal of the Optical Society of America*, 50(11):1139A, 1960.

[306] W. S. Stiles and J. M. Burch. "NPL Colour-Matching Investigation: Final Report," *Acta Optica*, 6:1–26, 1959.

[307] M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta. Standard default color space for the Internet, 1996, *www.w3.org/Graphics/Color/sRGB*.

[308] M. C. Stone. *A Field Guide to Digital Color*. Natick, MA: AK Peters, 2003.

[309] J. Stumpfel. "HDR Lighting Capture of the Sky and Sun," Master's thesis, California Institute of Technology, Pasadena, California, 2004.

[310] J. Stumpfel, A. Jones, A. Wenger, and P. Debevec. "Direct HDR Capture of the Sun and Sky," in *Proceedings of the 3rd International Conference on Virtual Reality, Computer Graphics, Visualization and Interaction in Africa (AFRIGRAPH 2004)*, 2004.

[311] L. Stupart. Dont Let the Sun Go Down on Me. Fluorescent paint on canvas, UV Light, 2008.

[312] K. Subr, C. Soler, and F. Durand. "Edge-Preserving Multiscale Image Decomposition Based on Local Extrema," *ACM Transactions on Graphics*, 28(5):147, 2009.

[313] S. Süsstrunk, J. Holm, and G. D. Finlayson. "Chromatic Adaptation Performance of Different RGB Sensors," in *Proceedings of IS&T/SPIE Electronic Imaging*, Vol. 4300, SPIE, 2001.

[314] J. Suzuki and I. Furukawa. "Required Number of Quantization Bits for CIE XYZ Signals Applied to Various Transforms in Digital Cinema Systems," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E90-A(5):1072–1084, 2007.

[315] R. Szeliski and H.-Y. Shum. "Creating Full View Panoramic Mosaics and Environment Maps," in *Proceedings of SIGGRAPH 97*, Computer Graphics Proceedings, Annual Conference Series, pp. 251–258, August 1997.

[316] E. Talvala, A. Adams, M. Horowitz, and M. Levoy. "Veiling Glare in High Dynamic Range Imaging," *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 26(3):37:1–9, 2007.

[317] C. Tchou and P. Debevec. "HDR Shop," 2001, *www.debevec.org/HDRShop*.

[318] C. Tchou, D. Maas, T. Hawkins, and P. Debevec. "Facial Reflectance Field Demo," SIGGRAPH 2000 Creative Applications Laboratory, 2000, *www.debevec.org/FaceDemo/*.

[319] P. C. Teo and D. J. Heeger. "Perceptual Image Distortion," in *IS&T/SPIE Conf on Human Vision, Visial Processing and Digital Display V*, Proceedings of the SPIE, Vol. 2179, pp. 127–139, 1994.

[320] P. C. Teo and D. J. Heeger. "A Model of Perceptual Image Fidelity," in *IEEE International Conference on Image Processing (ICIP)*, Vol. 2, pp. 2343, 1995.

[321] P. Thevenaz, U. E. Ruttimann, and M. Unser. "A Pyramid Approach to Subpixel registration based on intensity," *IEEE Transactions on Image Processing*, 7(1):27–41, 1998.

[322] C. Tomasi and R. Manduchi. "Bilateral Filtering for Gray and Color Images," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 836–846, 1998.

[323] M. Trentacoste. "Photometric Image Processing for High Dynamic Range Displays," Master's thesis, University of British Columbia, 2006.

[324] M. Trentacoste, W. Heidrich, L. Whitehead, H. Seetzen, and G. Ward. "Photometric Image Processing for High Dynamic Range Displays," *Journal of Visual Communication and Image Representation*, 18(5):439–451, 2007.

[325] J. Tumblin, J. K. Hodgins, and B. K. Guenter. "Two Methods for Display of High Contrast Images," *ACM Transactions on Graphics*, 18(1):56–94, 1999.

[326] J. Tumblin and H. Rushmeier. "Tone Reproduction for Computer Generated Images," *IEEE Computer Graphics and Applications*, 13(6):42–48, 1993.

[327] J. Tumblin and G. Turk. "LCIS: A Boundary Hierarchy for Detail-preserving Contrast Reduction," in A. Rockwood (ed.), *SIGGRAPH 1999, Computer Graphics Proceedings*, Annual Conference Series, pp. 83–90, Los Angeles, CA: Addison Wesley Longman, 1999.

[328] J. M. Valenton and D. Norrenvan Norren. "Light Adaptation of Primate Cones: An Analysis Based on Extracellular Data," *Vision Research*, 23:1539–1547, 1983.

[329] R. L. ValoisDe Valois, N. P. Cottaris, S. D. Elfar, L. E. Mahon, and J. A. Wilson. "Some Transformations of Color Information from Lateral Geniculate Nucleus to Striate Cortex," *Proceedings of the National Academy of Sciences of the United States of America*, 97(9):4997–5002, 2000.

[330] T. J. T. P. van den Berg, M. P. J. Hagenouw, and J. E. Coppens. "The Ciliary Corona: Physical Model and Simulation of the Fine Needles Radiating from

Point Light Sources," *Investigative Ophthalmology and Visual Science*, 46:2627–2632, 2005.

[331] A. Van Meeteren and J. J. Vos. "Resolution and Contrast Sensitivity at Low Luminances," *Vision Research*, 12:825–833, 1972.

[332] E. Veach and L. J. Guibas. "Metropolis Light Transport," in *Proceedings of SIGGRAPH 97*, Computer Graphics Proceedings, Annual Conference Series, pp. 65–76, 1997.

[333] J. Walraven and J. M. Valeton. "Visual Adaptation and Response Saturation," in A. J. Doornvan Doorn, W. A. Grindvan de Grind, and J. J. Koenderink (eds.), *Limits of Perception*, The Netherlands: VNU Press, pp. 401–429, 1984.

[334] B. A. Wandell. *Foundations of Vision*. Sinauer Associates, Inc., Sunderland, MA 1995.

[335] L. Wang, L. Wei, K. Zhou, B. Guo, and H.-Y. Shum. "High Dynamic Range Image Hallucination," in *18th Eurographics Symposium on Rendering*, pp. 321–326, 2007.

[336] Z. Wang, E. Simoncelli, and A. Bovik. "Multi-Scale Structural Similarity for Image Quality Assessment," *Proceedings of Signals, Systems and Computers*, 2:1398–1402, 2003.

[337] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[338] Z. Wang and A. C. Bovik. "A Universal Image Quality Index," *IEEE Signal Processing Letters*, 9(3):81–84, 2002.

[339] Z. Wang and A. C. Bovik. *Modern Image Quality Assessment*. San Rafael, CA, USA: Morgan & Claypool Publishers, 2006.

[340] G. Ward and M. Simmons. "Subband Encoding of High Dynamic Range Imagery," in *First ACM Symposium on Applied Perception in Graphics and Visualization (APGV)*, pp. 83–90, 2004.

[341] G. J. Ward. "Measuring and Modeling Anisotropic Reflection," *ACM Computer Graphics*, 26(2):265–272, 1992.

[342] G. Ward. "Real Pixels," in J. Arvo (ed.), *Graphics Gems II*, pp. 80–83, San Diego, CA, USA: Academic Press, 1992.

[343] G. Ward. "A Contrast-Based Scalefactor for Luminance Display," in P. Heckbert (ed.), *Graphics Gems IV*, pp. 415–421, Boston, MA: Academic Press, 1994.

[344] G. Ward. "A Wide Field, High Dynamic Range, Stereographic Viewer," in *Proceedings of PICS 2002*, 2002.

[345] G. Ward. "Fast, Robust Image Registration for Compositing High Dynamic Range Photographs from Hand-Held Exposures," *Journal of Graphics Tools*, 8(2):17–30, 2003.

[346] G. Ward and E. Eydelberg-Vileshin. "Picture Perfect RGB Rendering Using Spectral Prefiltering and Sharp Color Primaries," in P. Debevec and S. Gibson (eds.), *Thirteenth Eurographics Workshop on Rendering (2002)*, June 2002.

[347] G. Ward, H. Rushmeier, and C. Piatko. "A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes," *IEEE Transactions on Visualization and Computer Graphics*, 3(4):291–306, 1997.

[348] G. Ward and M. Simmons. "JPEG-HDR: A Backwards Compatible, High Dynamic Range Extension to JPEG," in *Proceedings of the Thirteenth Color Imaging Conference*, pp. 283–290, 2005.

[349] G. J. Ward. "The RADIANCE Lighting Simulation and Rendering System," in A. Glassner (ed.), *Proceedings of SIGGRAPH '94*, pp. 459–472, 1994.

[350] C. Ware. *Information Visualization: Perception for Design*. MA, USA: Morgan Kaufmann, 2000.

[351] A. B. Watson. "Efficiency of a Model Human Image Code," *Journal of the Optical Society A*, 12(4):2401–2417, 1987.

[352] A. B. Watson. "Temporal Sensitivity," in *Handbook of Perception and Human Performance*, Chapter 6, New York: Wiley, Wiley-Interscience, Chapter 6, pp. 1–43, 1986.

[353] A. B. Watson. "The Cortex Transform: Rapid Computation of Simulated Neural Images," *Computer Vision Graphics and Image Processing*, 39:311–327, 1987.

[354] A. B. Watson. "DCT Quantization Matrices Visually Optimized for Individual Images," in *Human Vision, Visual Processing, and Digital Display IV*, Vol. 1913–14, pp. 202–216, SPIE, 1993.

[355] A. B. Watson. "Visual Detection of Spatial Contrast Patterns: Evaluation of Five Simple Models," *Optics Express*, 6(1):12–33, 2000.

[356] A. B. Watson and J. A. Solomon. "A Model of Visual Contrast Gain Control and Pattern Masking," *Journal of the Optical Society of America A*, 14:2378–2390, 1997.

[357] B. Weiss. "Fast Median and Bilateral Filtering," *ACM Transactions on Graphics*, 25(3):519–526, 2006.

[358] G. Westheimer. "The Eye as an Optical Instrument," in K. R. Boff, L. Kaufman, and J.P. Thomas (eds.), *Handbook of Perception and Human Performance: 1. Sensory Processes and Perception*, pp. 4.1–4.20, New York: Wiley, 1986.

[359] P. Whittle. "Increments and Decrements: Luminance Discrimination," *Vision Research*, 26(10):1677–1691, 1986.

[360] T. Whitted. "An Improved Illumination Model for Shaded Display," *Communications of the ACM*, 23(6):343–349, June 1980.

[361] L. Williams. "Pyramidal Parametrics," *Computer Graphics (Proceedings of SIGGRAPH 83)*, 17(3):1–11, Detroit, MI, July, 1983.

[362] H. R. Wilson. "A Transducer Function for Threshold and Suprathreshold Human Vision," *Biological Cybernetics*, 38:171–178, 1980.

[363] H. R. Wilson. "Psychophysical Models of Spatial Vision and Hyperacuity," in D. Regan (ed.), *Spatial Vision, Vol. 10, Vision and Visual Disfunction*, pp. 179–206, Cambridge, MA: MIT Press, 1991.

[364] H. R. Wilson. "Psychophysical Models of Spatial Vision and Hyperacuity," in D. Regan (ed.), *Spatial Vision*, pp. 64–86, Boca Raton, FL: CRC Press, 1991.

[365] R. H. Wilson and J. Kim. "Dynamics of a Divisive Gain Control in Human Vision," *Vision Research*, 38:2735–2741, 1998.

[366] M. Winken, D. Marpe, H. Schwarz, and T. Wiegand. "Bit-Depth Scalable Video Coding," in *Proceedings of the International Conference on Image Processing*, pp. 5–8, 2007.

[367] S. Winkler. *Digital Video Quality: Vision Models and Metrics*. New York: John Wiley & Sons, 2005.

[368] S. Winkler. "Chapter 5: Perceptual Video Quality Metrics — A review," in H. R. Wu and K. R. Rao (eds.), *Digital Video Image Quality and Perceptual Coding*, pp. 155–179, Boca Raton, FL, USA: CRC Press, 2006.

[369] A. P. Witkin. "Scale-Space Filtering," in *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pp. 1019–1022, 1983.

[370] H. R. Wu and K. R. Rao. *Digital Video Image Quality and Perceptual Coding*. Boca Raton, FL: CRC Press, 2005.

[371] Y. Wu, Y. Gao, and Y. Chen. "Bit-Depth Scalability Compatible to H.264/AVC-Scalable Extension," *Journal of Visual Communication and Image Representation*, 19(6):372–381, 2008.

[372] G. Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*, 2nd ed., New York: John Wiley & Sons, 2000.

[373] Z. Xie and T. G. Stockham. "Towards the Unification of Three Visual Laws and Two Visual Models in Brightness Perception," *IEEE Transactions on Systems, Man, and Cybernetics*, 19:379–387, 1989.

[374] H. Yee and S. N. Pattanaik. "Segmentation and Adaptive Assimilation for Detail-preserving Display of High-Dynamic-Range Images," *The Visual Computer*, 19(7–8):457–466, 2003.

[375] A. Yoshida, M. Ihrke, R. Mantiuk, and H.-P. Seidel. "Brightness of Glare Illusion," in *Proceedings of ACM Symposium on Applied Perception in Graphics and Visualization*, pp. 83–90, Los Angeles, CA: ACM, 2008.

[376] A. Yoshida, R. Mantiuk, K. Myszkowski, and H.-P. Seidel. "Analysis of Reproducing Real-World Appearance on Displays of Varying Dynamic Range," *Computer Graphics Forum (Proceedings of EUROGRAPHICS 2006)*, 25(3):415–426, 2006.

[377] M. Yuen. "Chapter 3: Coding Artifacts and Visual Distrotions," in H. R. Wu and K. R. Rao (eds.), *Digital Video Image Quality and Perceptual Coding*, pp. 87–122, Boca Raton, FL, USA: CRC Press, 2006.

[378] D. Zavagno. "Some New Luminance-Gradient Effects," *Perception*, 28(7): 835–838, 1999.

[379] D. Zavagno and G. Caputo. "The Glare Effect and the Perception of Luminosity," *Perception*, 30(2):209–222, 2001.

[380] W. Zeng, S. Daly, and S. Lei. "Visual Optimization Tools in JPEG 2000," in *IEEE International Conference on Image Processing*, pp. 37–40, 2000.

[381] X. Zhang and B. A. Wandell. "A Spatial Extension of CIELAB for Digital Color Image Reproduction," *Society for Information Display Journal*, 5(1):61–63, 1997.

Index

16-bit/channel format, 102–103
24-bit RGB image, 91

Adaptive countershading, 360–363
Adaptive filtering, 423
Adaptive gain control, 294–297
Adobe RGB color space, 86
Ambient occlusion, 583–586
Amplitude nonlinearity modeling, 460–464
Analogous threshold-versus-intensity (TVI) function, 461
Analysis-synthesis filter banks, 342
Angular map, 537–538
Animation, 504
– light probe image
– – acquiring and assembling, 504–506
– – mapping, 506–508
– postprocessing renderings, 508–514
– RNL, 543, 547
Apparent contrast enhancement, 348
Appearance correlates, 64
– computation of perceptual, 70–71
Applications of HDR images, 93–95
Ashikhmin's operator, 332, 334–338

Background intensity
– image average, 256–257
– local average, 257–260
– multiscale adaptation, 260–262
Backwards-compatibility, 120
– video compression, 136–144
Backward models, 270–276
Band filters, 482
Band-pass filters, 339
Bidirectional reflectance distribution functions (BRDFs), 24, 547
Bilateral filter, 293–295, 312, 380, 381
– computation time of, 387, 389
– smoothed image, 384
Bilateral filtering, 259, 379, 380, 386, 388, 390
– HDR image tone mapped with, 384, 385
Bit-depth expansion (BDE) technique, 423
Bitmap images, 156
BitmapShift function, 166
BitmapShift operator, 167
BitmapTotal function, 166
BitmapTotal operator, 167
BitmapXOR function, 166
Blackbody radiator, 37
Bleaching, 243
Blind spots, 517
Bloom effect, 459, 510
Bradford chromatic adaptation transform, 40
BRDFs, *see* Bidirectional reflectance distribution functions
Brightness
– definition of, 64
– encoding, 80–82
Brightness enhancement function, 348–351, 413, 414
Brightness-preserving operator, Tumblin–Rushmeier, 324–329
Brute force method, 159

© 2010, Elsevier Inc. All rights reserved.
DOI: 10.1016/B978-0-12-374914-7.00014-2

Calibration, 265–268
Calligraphic mode, 208
Camera, 145
– Genesis, 201
– point spread function, 190–193
– response function, 158, 171
– – by caveats and calibration, 182–183
– – by Debevec and Malik technique, 171–174
– – by Mitsunaga and Nayar technique, 174–177
– response recovery, 177–180
– Viper FilmStream, 200
Camera RAW, 147
Camera response recovery, 177–180
CAT02 chromatic adaptation, 43, 44
Cathode ray tube (CRT), 5, 206–208
Center-surround mechanisms, 289, 300
Chroma, definition of, 65
Chromatic adaptation, 39–44, 46, 67–68
– CAT02, 43, 44
Chromaticity
– coordinates, 31, 32, 34
– diagrams for color spaces, 89
– for RGB color spaces, 85
CIE, *see* Commission Internationale de l'Eclairage
CIECAM02 color appearance model, 66–72, 306
– refinements, 72–77
CIELAB color space, 58–59
CIELUV color space, 56–57
Cineon format, 104
Civetta, 202
Clamping technique, 393, 394
Clipped regions, 418
– process of enhancement, 420
Cold cathode fluorescent lamps (CCFLs), 209
Color appearance, 61–66
– CIECAM02 model, 66–72
– models, 297, 306
– refinements, 72–77
– under extended luminance levels, 317–323
Color coding, 486
Color constancy, 39, 48
Color contrast, simultaneous, 63, 64
Color correction, 61
Color gamut, 33, 48
Color images, 268–269
Color opponent spaces, 48–53
– CIELAB, 58–59
– CIELUV, 56–57
– IPT, 59–60
Color spaces, 34–36
– Adobe RGB, 86
– chromaticity diagrams for, 89
– Hunt–Pointer–Estevez, 68
– for lossy encoding, 123–130
– sRGB, 33, 84, 89, 443
– standard RGB, 82–88
Color temperature, 38
– correlated, 38
Color-matching functions, 28–31
Colorimetry, 28–33
Commission Internationale de l'Eclairage (CIE), 26, 30, 32
– D_{55}, D_{65}, D_{75} illuminants, 36, 37, 39
– 1931 Standard Observer, 30, 33
– XYZ color-matching functions, 31
Companding, 423
Complementary metal oxide semiconductor (CMOS) image sensor, 201
ComputeBitmaps function, 166
Computer games, 94
Cone cells, 240–243, 251, 253, 254
– photopigment, 243
– response curve of, 243–245
Consumer level photo cameras, 12
Contouring for chromatic channels, 422
Contrast adaptation, 453
Contrast comparison function, 449
Contrast detection threshold, 494
Contrast discrimination threshold, 474, 489
Contrast distortion, 493
Contrast equalization, 397, 398
Contrast increments, 488
Contrast mapping, 397, 398

Contrast perception, 352
Contrast sensitivity function (CSF), 453, 464–468
– factors influencing, 464
– modeling, 470
Contrast sensitivity processing, 468
Contrast-versus-intensity (CVI) function, 461, 462
– curve, 494, 495
Coring technique, 423
Cornsweet illusion, 351
– and physical contrast, 352
– apparent contrast effects, 353, 354, 357, 358
Cornsweet profiles, 354
– amplitude of, 358, 363
– cascading, 356, 357
– into achromatic channel, 365
– into chroma channel, 364
Correlated color temperature, 37, 38
Cortex transform, 468–472
Cortex transform filters, 482
Countershading, 355, 360, 361, 363
CRT, *see* Cathode ray tube
Cube map, 540–542
Cumulative distribution function (CDF), 569
Custom video coding, 121, 123

Dark adaptation, 263
Denormalization, floating point, 82
Detection contrast threshold, 472, 495
Difference of Gaussians (DoGs), 289, 300, 339, 345
Digital camera, optical limitations, 188
Digital cinema, 94, 129
Digital color encoding, 81
Digital light-processing (DLP) projectors, 215
Digital photography, 93
Digital single-lens reflex (DSLR), 147
Disability glare, 367
Discomfort glare, 367
Discrete cosine transforms (DCT), 132
Discrimination threshold, 472, 474, 487
Display-adaptive tone reproduction, 343–348
– control flow of, 344
– for display conditions, 349, 350
Display gamma, 77–80
Distortion measure image, 454
DoGs, *see* Difference of Gaussians
Dolby prototype HDR display, 11
Drago's adaptive logarithmic mapping, 484
Drago's operator, 485
Dry developing method, 147
Dual modulation displays, 216–218
– in large screen, 220–221
Dual modulation screens, image processing for, 221–224
Dynamic range, 4
Dynamic range expansion functions, 406
– expanding highlights, 406–408
– gamma expansion, 410–412
Dynamic range–independent (DRI) image-quality metric, 479–483
– accounting for maladaptation, 497
– applications, 483–486
– comparing with SSIM and HDR VDP, 486
Dynamic range reduction, 14
Dynamics of visual adaptation, 262–264

Edge bitmaps, 157, 158
Edge-matching algorithms, 156
Electronic paper, 213
Electrophoretic displays, 213
Emerging display technologies, 215–216
Encoding, 91
– brightness, 80–82
– color space for, 123–130
– comparison, 110–115
– error levels for, 114
– logarithmic, 95
– LogLuv, 98–102, 111
– lossy, 105–109
– low versus high dynamic range, 91–92

– quantization and, 422–424
– video, *see* Video encoding
Environment mapping, 503, 579–583
Equirectangular mapping, 540
Exclusion bitmap, 162, 163
Exclusive-or (XOR) difference, 163
– computing, 161
Exposure fusion, 399–404
EXtended Range format (.exr), 103
Eye contrast sensitivity, 461
Eye optics
– modeling, 458–460
– simulation, 456

Fast Fourier transform (FFT), 374
Fattal's gradient domain compression, 484
Fattal's operator, 485
Fechner's law, 461, 462, 488
Fiat Lux animation, 557, 559, 560, 577, 578
Field sequential color, 214
Film camera, 145
Film emulsions, response curves, 146
Film scanning, 151
FilmStream Viper camera, 12
Fish-eye lenses, 523
Floating point, 82
– denormalization, 82
Forward adaptation model, 252, 270–276
Fourier transforms, 305, 372
Frame averaging, 183, 184
Frames, 138–140

Gain control function, 253–254
Gamma correction, 410–412
Gamma encoding, 80–82
– advantages of, 80
Gamma estimation for CRT displays, 78
Gamut, color, 33, 48
Gaussian blur, 414
Gaussian filters, 292, 295, 329, 336, 370, 381
Gelb effect, 339
Genesis, 201
GetExpShift algorithm, 164, 165
Ghost removal, 185–187
Glare effects, 459
– ciliary corona, 371
– in eye, 367, 369
– lenticular halo, 372
Glare illusion, 367
– computation process, 373, 374
– wave-optics model, 371, 372
Glare pattern for light source, 368
Global illumination
– algorithms, 503
– computation, 543–549
Gradient domain compression, 390–395
Grating orientation, 464

Hallucination, 416–418
Hat function, 151
HDTV color space, 86, 89
Hemispherical fisheye projection, 229, 230
Heuristic image-processing algorithm, 223
High dynamic range (HDR) format, 96–98
Histogram adjustment operator, 376–380, 382, 383
Histograms
– cumulative, 181
– deriving sample values from, 181–182
Hue
– angles, 70, 71
– definition of, 64
Human visual sensitivity, 457
Human visual system (HVS), 236, 237, 268, 270, 271, 278, 440
– adaptation, 237–238
– models based on, 323
– fidelity metrics, 451, 452
– photoreceptor mechanisms, 243–251
– pupil, 239–240
– rod and cone cells, 240–242
Hunt effect, 280, 316, 319
Hunt–Pointer–Estevez color space, 68
HVS, *see* Human visual system

IBL, *see* Image-based lighting
iCAM model, 72, 306–312
– chromatic adaptation transform, 307
– exponential function, 308, 309
– prescaling effect on, 310
iCAM06 model, 312–317, 440

Ideal mirrored sphere mapping, 535–537
Illuminance, 27
Illuminants, 36–47
Image alignment technique, 155
Image capture from multiple exposures, 148–151
Image distortion, 484
– measure, 454
Image editing, 94
Image file format, 95
– high dynamic range, 96–98
– OpenEXR, 103–104, 111
– tagged image file format, 98–102
Image fusion, 399
Image pyramid, 159
Image registration techniques, 154
Image resolution, 520
Image-based lighting (IBL), 501
– application of, 503
– approximations, 579–586
– for real objects and people, 586–593
– global illumination computation, 543–549
– postprocessing for, 508–514
– renderings of 3D objects, 592
– sampling incident illumination, 549–553
Image-based material editing, 433
ImageShrink2 function, 166
Importance sampling, 569–572
In-plane switching (IPS), 211, 212
Interchannel masking, 477
International Electrotechnical Commission (IEC), 84
International Telecommunication Union (ITU)
– recommendation BT.709, 33, 35, 36
Intrachannel contrast, classifying, 480
Intrachannel masking, 476
Inverse adaptation model, 252
Inverse fast Fourier transform (IFFT), 374
Inverse tone mapping, 143, 480
Inverse tone-mapping operators (iTMOs), 479
Inverse tone reproduction, 405, 407
– applications, 432–434
– inverse photographic operator, 408–410
Inverted system response function, 152
Invisible (subthreshold) distortion, 458
Invisible contrast
– amplification of, 480
– application of, 480
IPS, *see* In-plane switching
IPT color space, 59–60
Irradiance, 20, 21
– caching, 547

JND, *see* Just noticeable difference
JPEG-HDR format, 105–108
Just noticeable difference (JND), 237, 252, 256
– effect of, 466
– luma encoding, 445
– scaled space, 444–447

K-means clustering process, 565, 566

Laplacian filter, 401
Large expanse extra perspective (LEEP) ARV-1 optics, 229
Laser displays, 216
Laser speckle, 216
Latitude–longitude mapping, 538–540
LCDs, *see* Liquid crystal displays
LCoS, *see* Liquid-crystal-on-silicon projectors
LDR image, *see* Low dynamic range image
LDR2HDR method, 412–415, 430
Legge–Foley transducer function, 488
Lens flare, 366
– removal, 191
Light
– adaptation, 263
– colorimetry and measuring, 28–33

– photometry and measuring, 25–28
– radiometry and measuring, 19–25
Light-emitting diodes (LEDs), 6, 212
Light probe, converting to light sources, 561–568
Light probe image, 503
– acquiring and assembling, 504–506
– capturing, 515
– – environments with bright sources, 525–534
– – fish-eye lenses, 523
– – mirrored sphere, photographing, 516–520
– – scanning panoramic cameras, 523–525
– – tiled photographs, 520–521
– mapping, 506–508
Light sources
– constellation, 561–568
– identification, 553–559
LightGen plug-in, 565, 566
Lightness
– definition of, 64
– perception, 338–339
Linear scaling method, 427–428
Liquid crystal displays (LCDs), 209–213
Liquid-crystal-on-silicon (LCoS) projectors, 215
Local-dimming HDR displays, 216–224
Local-dimming television screens, 219–220
Local scene rendering into nondiffuse, 577–578
Log encoding, 81, 82
Logarithmic encoding, 95, 443
Logarithmic inverse response function, 172, 173
LogLuv encoding, 98–102, 111
Lossy video compression, 120
Low dynamic range (LDR) image, 4, 285
– correcting white balance, 7
– encoding, 91–92
– expanding highlights in, 406–408
– frames, 138
– using inverse photographic operator, 408–410
Luma encoding, 444
Luma values, LDR and HDR, 140
Luminance, 27, 28, 442
– comparison function, 449
– encoding, 445
– increments, 488
– luma mapping in sRGB, 444
– masking, 457, 461
– quantization error, 444
– thresholds, visible, 444
Luminance-domain filter kernel, 295, 296
Luminous exitance, 27
Luminous intensity, 27
Luminous power/flux, 27
Macbeth ColorChecker^{tm} chart, 62
Mach bands, 352, 353
Mantiuk's transducer, 491–492
– disadvantage of, 492
Masker contrast, 474
Mean square error (MSE)
– advantages, 446
– minimization of, 441
Median threshold bitmap (MTB) alignment, 155–159, 168
– adding rotational alignment to, 169
– close-up of pixels in, 161
– computing, 170
– efficiency of, 167
– features, 156
– overall algorithm, 164–167
– pyramid of, 160
– threshold noise, 161–164
Metamerism, 33
Michelson contrast, 491
Micro-mirror projectors, 215
Microsoft's HD photo format, 110
Mirror-like specular surface, 545–546
Mirrored sphere mapping, 535–537
Mirrored spheres, photographing, 516
– blind spots, 517
– calibrating sphere reflectivity, 517–518
– framing and focus, 516
– nonspecular reflectance, 518
– polarized reflectance, 519

Mitsunaga–Nayar weighting function, 152
Motion vector, 154
MPEG-4 video encoding, 130–133
– backwards-compatibility of, 137–140
MTB alignment, *see* Median threshold bitmap alignment
Multi-scale image decompositions, 398, 399
Multichannel decomposition technique, 468
Multichannel image-quality metrics, 455
Multichannel models, 453–454
Multidomain vertical alignment (MVA) design, 211, 212
Multiple exposures, image capture of, 148, 151, 198–199
Multiresolution edge-detection scheme, 391
Multiscale contrast processing, 396
Multiscale observer model, 297, 306
– filter kernel size, 303
– forward model, 298–300
– interpolation parameter, 301
– inverse model, 300
– prescaling effect on, 305
– relative scaling in, 304
Multiscale optimization frameworks, 395–399

Naka–Rushton equation, 245–247, 263, 292
Neutral density (ND) filter, 526
Noise removal, 183–185
Nonlinear filtering, processing flow, 468
Nonlinear response compression, 69
Normalized contrast sensitivity function (nCSF), 467
NTSC color space, 89

Objective metrics, 437
– full-reference (FR), 438–440
– pixel-based, 440–441
OLED, *see* Organic light-emitting diode display
Omnidirectional image mapping, 535
Omnidirectional photography, 503
Organic light-emitting diode (OLED) displays, 215–216
OpenEXR format, 103–104, 111
Optical light scattering, effects of, 466
Optical transfer function (OTF), 369, 456, 459, 460
Out-of-gamut, 33
Output-referred standards, 83

Panavision, Genesis, 201
Parthenon model, 550, 552, 554, 557
Peak signal-to-noise ratio (PSNR) metric, 441
Percentile threshold bitmap, 158
Perception-based fidelity metrics, 451–455
– multichannel image-quality metrics, 454–455
– multichannel models, 453–454
Perception-based metrics, 439
Perceptual effects, 349
Phase uncertainty, 476
Photochemical reaction, 243
Photographic dodging-and-burning, 285, 287, 291
Photographic tone-mapping function, 255
Photographic tone-reproduction operator, 285–291
– L_{white} parameter in, 288
– prescaling image data, 286, 287
– sharpening parameter Φ in, 290
Photometry, 25–28
– quantities of, 27
Photon shot noise, 184
Photons, 23
Photopigment, 263
– depletion and regeneration, 243

Photoreceptor, 240, 243
– adaptation, 245–247, 251
– – model for tone mapping, 255
– mechanisms, 243–245
– types of, 240
Photoreceptor model, 277
– chromatic adaptation in, 281, 282
– light adaptation in, 283
– luminance control with, 279
– luminance mapping by, 284
Physically based rendering, 93
Pixar log encoding, 104
Pixel encoding, 123–130
Pixel intensity encoding, perceptually uniform spaces for, 442–443
Pixel intensity saturation, 485
Pixel luminance, 451
Pixel-based metrics, 439
Pixels, 152, 162
– reference exposure, 187
Pixim, CMOS image sensor, 201–202
Planckian locus, 39
Plasma displays, 208–209
Point spread function (PSF), 190–193, 367, 372, 374
– estimating, 193–195
– removing, 196–197
Posterization, 422
Power law, 77, 80, 81
Predictive cancellation technique, 424
Primary stimuli, 28
Principal components analysis (PCA), 51
Printing, 224–225
– reflective, 225–227
Probability distribution function (PDF), 569
Probe mapping, 544
Projection technologies, 214–215
Projector-integrated second modulator, 220
PSF, *see* Point spread function
Psychometric function, 477
Pupil, 239–240

Quality metrics
– classification, 438
– subjective versus objective, 436–437
Quantization
– and encoding, 422–424
– errors, 127, 128
– matrices, 132
QuicktimeVR^{tm}, 94

Radiance, 23
Radiance picture format, *see* High dynamic range format
Radiant energy, 19, 20
Radiant exitance, 20, 21
Radiant flux density, 20
Radiant intensity, 20, 22
Radiant power, 20
Radiometry, 19–25
– quantities of, 20
Rational quantization function, 253
Reference exposure, 186, 187
Reflectance, 19
– surface, types, 547–549
Reflection occlusion, 581
Reflection print, 225–227
Reflective display technologies, 213–214
Reflective LCD panels, 213–214
Reinhard's operator, 485
Reinhard's photographic tone reproduction, 484
Relative discrimination threshold, 475
Relative exposure ratios, 171
Relighting, 588–593
Remote sensing, 93
Rendering
– differential, 576–577
– into nondiffuse local scene, 577–578
Rendering with natural light (RNL), 504
– animation, 543, 547
– renderings for, 511
Response–threshold relation, 247–251
Retinex model, 329–333
RGB color cube, 33
RGB color spaces
– Adobe, 86
– chromaticity for, 85
– sRGB, 33, 84, 89, 443
– standard, 82–89
– transformations for, 87–88
RNL, *see* Rendering with natural light
Rod cells, 240–243, 251, 253, 254
– photopigment, 243
– response curve of, 243–245

S-shaped curve, 254
Sampling incident illumination, 549–553
Sampling problem, 552
Saturation, definition of, 65
Scalable bit-depth coding, 122–123
– in H.264/AVC framework, 141–144
Scalable video coding (SVC), 120
Scanning panoramic cameras, 523–525
Scene–object interreflection, 572–576
Scene-referred standard, 84
Screen modulation approach, 221
Sensor noise, 148
Shadows simulation, 572–576
SIGGRAPH 99 electronic theater animation, 557
Sigmoidal compression, 313
– local, 291–294
Sigmoidal functions, 277, 278, 280
Sigmoidal tone-reproduction operators, 277
Signal theory, 151
Simulcast, 122
SMPTE-240M color space, 86
Spatial filter kernel, 295, 296
Spatial frequency aftereffects, 453
Spatial masking, *see* Visual masking
Spectral sharpening, 47–48
SpheroCam HDR image, 202
SpheronVR, 202
sRGB color space, 33, 84, 89, 443
Stevens effect, 316, 319
Stevens' power law, 461, 462, 488
Structural similarity (SSIM) index, 448–451
– sensitivity, 479
Structure comparison function, 450
Structure-based metrics, 439
Subband encoding, 339–343
Subthreshold summation, 453
Suprathreshold distortion, 452
Suprathreshold image-quality metrics, 487–489
– accounting for maladaptation, 495
Suprathreshold magnitude estimation, 445
Surface reflectance types, 547–549

Tagged image file format, 98–103
Texture synthesis technique, 415, 416
Threshold bitmaps, 159
Threshold contrast, 494
Threshold elevation maps, 457
Threshold elevation model, 474
Threshold fidelity metric, 458
Threshold noise, 161–164
Threshold-versus-intensity (TVI), 238
– function, 126
– model for tone mapping, 255–256
Tiled photographs, 520–521
Tone mapping, 92, 112, 483, 484, 508
– background intensity, 256–262
– forward, 142
– photographic function, 255
– photoreceptor adaptation model for, 252–255
– problem, 233–237
– TVI model for, 255–256
– visual adaptation models for, 252–256
– Ward's algorithm, 255
Tone-mapping operators, comparison of, 485
Tone-reproduction operators, 264, 270
– calibration, 265–268
– color images, 268–269
– forward and backward models, 270–276
– HVS-based models, 323
– image appearance models, 297
– sigmoidal, 277
Transducer function, 489, 490
Transflective displays, 213
Transparent media, 227–228
Trilateral filtering, 260
TVI, *see* Threshold-versus-intensity

Twisted nematic (TN) cell design, 210, 211

Under- and overexposed image
– dynamic range expansion, 428–431
– reconstruction of clipped regions in, 415–420
Unsharp masking, 359, 365, 366

Veiling glare, 218, 366, 379
Video encoding
– color space for lossy pixel, 123–130
– interlayer prediction for, 142
– MPEG-4 framework, 130–133
– strategies, 121–123
Video player, 133–136
Video processing, 418–420
Vignetting, 514
Viper FilmStream, 200–201
– response curve, 201
Virtual reality, 94
Visibility cache, 572
Visible (suprathreshold) distortion, 458
Visible contrast
– loss of, 480
– reversal of, 480
Visible differences predictor (VDP), 454–458
– amplitude nonlinearity modeling, 460–464
– cortex transform, 468–472
– CSF, 464–468
– eye optics modeling, 458–460
– phase uncertainty, 476
– probability summation, 477
– psychometric function, 477
– visual masking, 472–477
Visual adaptation
– dynamics of, 262–264
– models for tone mapping, 252–256
Visual characteristics, 453
Visual masking, 453, 457, 472
Visual threshold, 237
von Kries chromatic adaptation transform, 307

Ward's HDR still viewer, 228–230
Ward's tone-mapping algorithm, 255
Weighted least squares (WLS) multiscale method, 398–400
Weighting function, 151, 173
Weiss AG, 202
Well-exposed images, 425–428
White balancing, 37, 62
White point, 36–47
Wide gamut color space, 89
Wilson's transducer, 489–491, 493
– metric based on, 493–494

XDepth format, 108
XYZ color space, 34
– color-matching functions, 31
– converting from RGB to, 34

Zone system, 285

This page intentionally left blank

This page intentionally left blank

This page intentionally left blank

This page intentionally left blank